UIUCDCS-R-73-556

COO-1469-0216

APE MACHINE

A NOVEL STOCHASTIC COMPUTER BASED ON A SET OF AUTOMONOUS PROCESSING ELEMENTS

by

YIU KWAN WO

February, 1973

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

UIUCDCS-R-73-556

APE MACHINE
A NOVEL STOCHASTIC COMPUTER BASED ON A SET OF AUTOMONOUS PROCESSING ELEMENTS

by

YIU KWAN WO

February, 1973

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

* This work was supported in part by Contract No. US AEC AT(11-1)1469 and was submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering, February, 1973.

ACKNOWLEDGEMENT

The author wishes to express his sincerest gratitude to his advisor, Professor W. J. Poppelbaum, for suggesting this thesis topic and for his continuous guidance, friendship, and invaluable support throughout the entire course of his graduate study. He is also grateful to his co-worker D. L. Olson, who built part of the APE machine.

Furthermore, he would like to thank all other members of the Hardware and Systems Research Group for their friendship and many valued discussions over the past few years. He is also indebted to all the personnel in the fabrication group and machine shop under Frank Serio for the excellent work they contributed toward the building of the APE machine. Special thanks go to Jerry Fiscus, who did an outstanding job in some difficult high density circuit layouts.

All of the part-time technicians, Lloyd Ravlin, Mike Selander, Jerry Menchhoff, and Bert Semmelink who worked on the APE project at various times during the past two years are thanked for their help.

Thanks are also due to Barbara Bunting for typing this thesis, to Mark Goebel for the drawings, and to Bernard Tse for typing the first draft.

Last, but not least, he would like to thank his wife, Catherine, for her support, understanding, and encouragement throughout the project.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Rapid development in Integrated Circuit Technology together with the increasingly widespread use of computers in recent years has brought about many significant changes in computer design philosophy. ICs now can be mass-produced at very low cost: The resulting drop in the relative cost, as compared to that of other components, other construction and operation costs of a computer, leads to a new set of criteria in determining an optimum system organization and the design of the detail circuits. These new criteria inevitably emphasize on extensive use of mass-producible items like the LSI and avoid as much as possible the components which are not mass-producible and construction processes which are difficult to automate. On the other hand, extensive use of computer by people from all walks of life calls for new types of computers with more flexible system organization, higher reliability and simpler operation procedures.

Under these new conditions, a novel stochastic computer with highly flexible structure, originally suggested by Professor Poppelbaum, is being built. It takes advantage of the latest advances in component technology, and leads to new standards of computer structure and performance. This computer is dubbed APE machine[1] as its basic building blocks are a set of Autonomous Processing Elements. These basic building blocks consist of absolutely identical circuits (except for transmitter frequencies), making the structure of the APE machine highly homogeneous. This homogeneous structure lends itself exceedingly well to mass-production by integrated circuit technology.

Each basic building block is a small processor in its own right. However they can be grouped together to form a more powerful system, without any actual physical hook-ups. The freedom and ease of incrementing its computing power by simply adding more APEs makes the APE machine a futuristic design in keeping with the changing needs of computer users: At the present time, an increase in the computing power of a computer system usually requires the replacement of the system or a major processing unit. (For a small but increasing number of recent models, provision is made to allow additional processors to be connected to the existing facilities to boost the computing capability. But in all cases, the user has no control over the size of the incremental power and has little influence in organizing the computer system in a problem-dependent way in order to obtain better performances.) For the APE machine, a bigger computing capability simply means using more basic building blocks, and these building blocks can be organized in various ways to suit particular applications!

Fault tolerance is a highly desirable feature for complicated machines such as computers. It is even more so for computers handling critical jobs. Research in this area[2][3] has grown rapidly over the past decade. It is quite clear at this point that fault tolerance is going to be one of the most important features of future computers. Should the need arise, additional circuitry could be easily added to the APE machine such that various fault tolerant features (such as static self-checking, dynamic self-checking, and even self-repair) may be incorporated.

The processing method of the APE machine is based on the stochastic processing principles to be discussed in Chapter 3. Stochastic processing is

used because the project APE machine represents an effort to explore the new approaches to data processing and to develop new processing techniques and new hardware. Stochastic processing has been investigated both in the U.S. and abroad[4][5][6][7][8]. Research conducted in the Hardware and Systems Research Group under Professor Poppelbaum has made significant contribution to the theory and practice in this area. Several complex computing machines based on stochastic computation principles have been successfully constructed over the past few years by this research group[9][10][11]. The APE machine is yet another attempt to further investigate this area.

In the following chapters, the system structure and the design of the APE machine will be discussed. Some detail analysis is given to its stochastic operations. Its limitations as well as their possible solutions are also presented.

2. SYSTEM DESCRIPTION OF THE APE MACHINE

2.1 General Description

The APE machine is an on-line real-time stochastic computer with reconfigurable structure. It consists of a set of Autonomous Processing Elements known as APEs, a set of sensors to acquire raw input data for processing, a remote power supply and a program control unit. The system diagram of the APE machine is depicted in Figure 2.1. Data communication between various constituents of the APE machine is carried out through radio frequency channels only. Power is also transmitted remotely into the APEs in the form of light energy. Therefore the APE machine does not require assembly into a physically wired network. For communication with other parts of the computer each APE has two tunable data input channels, a fixed instruction channel, and an output transmitting channel operated at the same fixed frequency but on a time-multiplexed basis with the instruction channel. The two data input channels can be tuned to establish RF linkages with the output of any other APEs or sensors. The APE can perform on its two input data any of the operations of addition, subtraction, multiplication, division, differentiation, integration and storage. The tuning of the data input channels as well as the setting for an APE to perform a specific operation is done remotely through the instruction channel linking with the control unit. The sensors provide the system interface with the outside world: They convert the input variables, such as temperature, light intensity, etc., into machine variables compatible with the data input channels of the APEs.

In Figure 2.1, the APEs are symbolized by triangles with i and j denoting the tunable data input channels and k denoting the fixed output channel. A specific output channel k is identified by its channel frequency

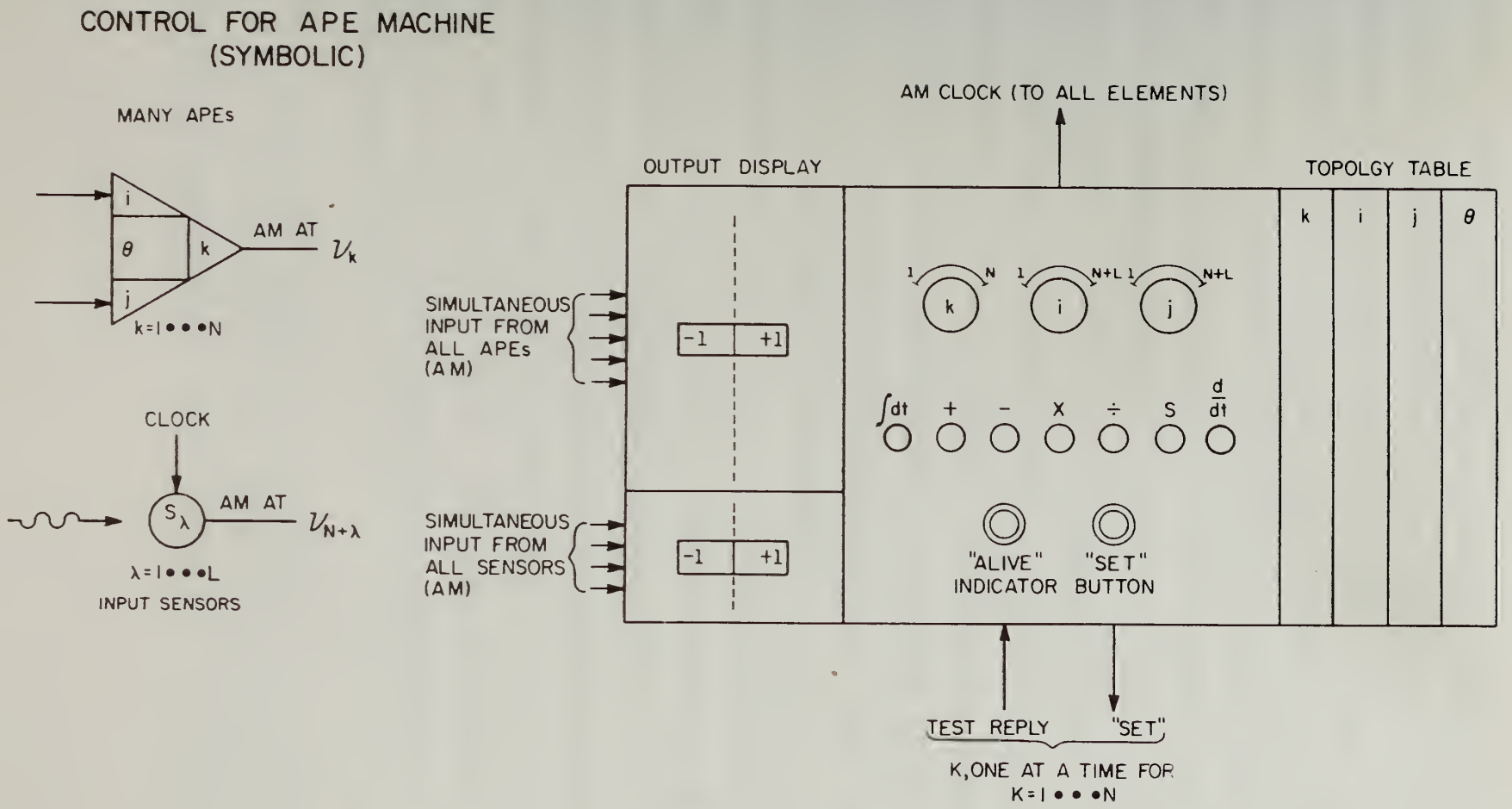


Figure 2.1 The APE Machine

ν_k. There are N number of different output channels available to the system. As mentioned before the computational power of the APE machine is incrementable. However, the increment is not without bound. The number N is a parameter indicating the maximum computational power which a particular APE machine can obtain. It should be noted that the number of APEs in a specific APE machine might be greater or smaller than N. In other words, it is allowed to have redundant APEs operating in parallel or completely missing APEs for any particular channel. This will be discussed in more detail later on. A specific operation instruction for an APE is denoted by θ. It could be any one of the seven operations mentioned above. The sensors are denoted by a circle in Figure 2.1. Their outputs are transmitted through channels other than those for the APEs. Their channel frequencies are denoted by $\nu_{N+\lambda}$.

A simple example of organizing the APEs and sensors to perform $a \cdot b + c^2$ is illustrated in Figure 2.2. In this example, three input variables a, b and c are provided by sensors S_1, S_2 and S_3. Their values are encoded and transmitted from their respective output channels. The input channels i and j of the APEs belonging to the channels denoted by l and m are tuned to receive input data from S_1, S_2 and S_3 as shown. APEs in channels l and m are set to perform multiplication. The APE or APEs operating in the channel denoted by n, with their data input channels tuned to receive input data from the output of APE channel l and APE channel m, are set to perform addition. The final result $a \cdot b + c^2$ is then transmitted at frequency ν_n. The program control unit is equipped with an all-channel receiver. It can be set to receive the final result $a \cdot b + c^2$ from APE channel n as well as outputs from any

other APE channels and sensor channels. The display of the result is done by means of a Nixie-tube display panel. For more permanent outputs, provision is made to interface with a teletype for printing the outputs.

Programming of the APE machine can be carried out by selecting two APE channels i and j for the two tunable data input receivers and assigning a specific operation θ for each of the APEs involved, as symbolized by the i, j and k dials together with a row of buttons for different operations in Figure 2.1. The clock is used for synchronization while the 'Alive' and 'Set' buttons indicate part of the fault tolerant features and are used to test an APE channel before it is put into operation. More detail about synchronization and testing will be discussed later. The topology table of the control unit shown in the figure simply serves symbolically as a convenient place to keep a record of the program of the APE machine.

The APE machine has three different modes of operation. In the test mode, test signals are sent out to check all APE channels involved in the program to be executed. In the programming mode, the APEs and sensors are organized in a suitable structure according to the program. During the execution mode, data transmission and data processing are done by the APEs in a synchronous manner. All APEs communicate with each other during the communication period of the computing cycle and process the data during the remaining part of the computing cycle.

2.2 The Modulation Scheme Employed by the APE Machine for Data Communication

Special consideration is required in choosing the proper modulation scheme so that more than one APE having identical output channel frequency

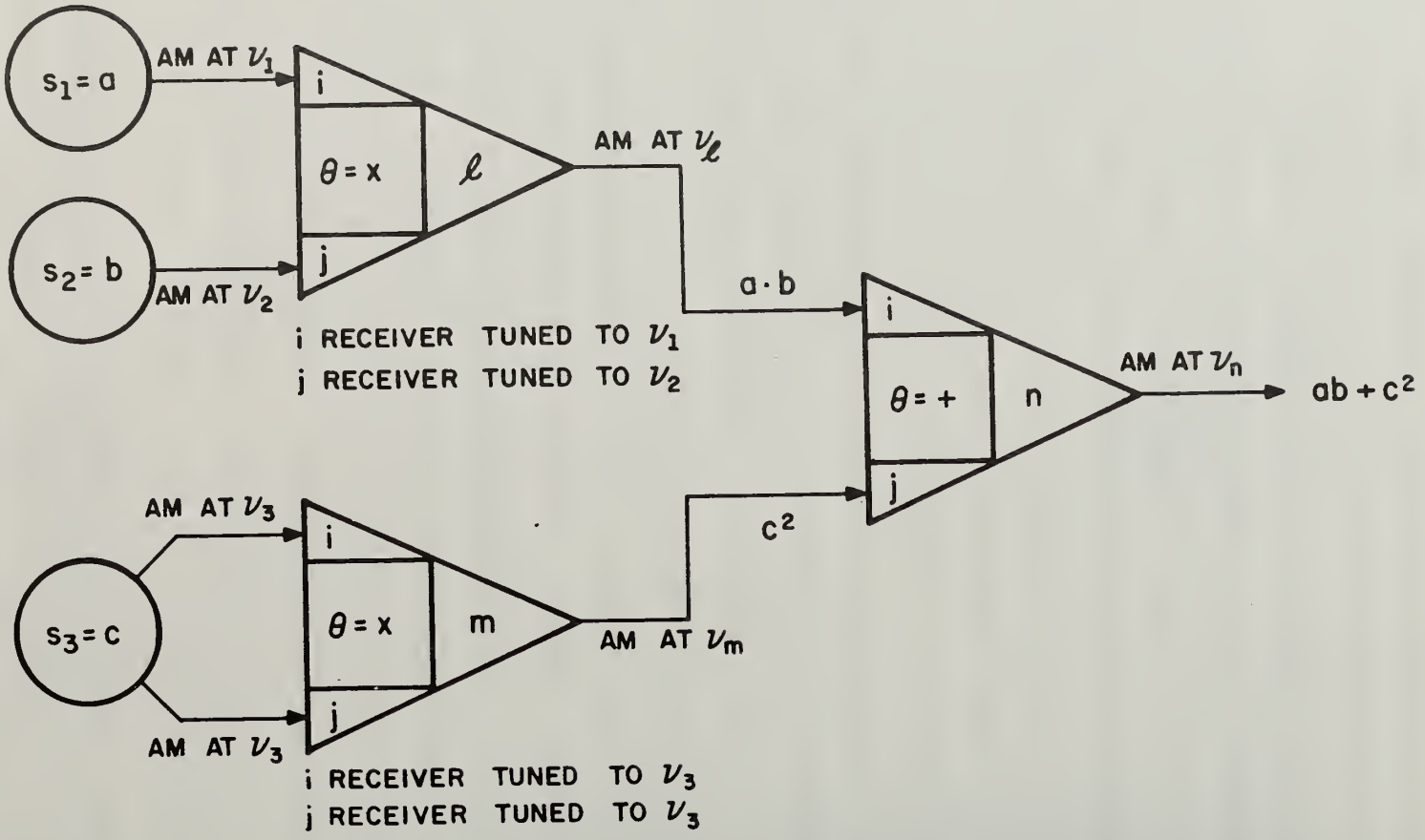


Figure 2.2 A Simple Program to Compute $a \cdot b + c^2$

are allowed. In the case of several APEs having the same output channel, they are said to be of the same type. All APEs of the same type operate in parallel in every respect. Their corresponding input data channels are tuned to receive data from the same source, and they are set to perform the same operation. In other words, the set of APEs actually consists of many different types of APEs. A specific type of APE has the same specific fixed frequency channel for instruction input and its data output. Whenever an instruction is sent out from the program control unit, all APEs of the same type will be programmed exactly the same way. To guarantee a proper reception of data from the same type of APEs by the APEs in the following stage, a synchronous pulse width modulation scheme is employed for data communication. All APEs and sensors are synchronized for data transmission and reception: At exactly the same time, each of the APEs and sensors begins to transmit an RF pulse whose width carries the information. If there are several APEs in one particular channel, they stop transmission at about the same time due to the fact that the data sent out from each of them are equal (to within the accuracy of the processor of the APEs). With this kind of modulation scheme, it is evident that the data transmission is not seriously degraded by the existence of redundant APEs.

2.3 Flexibility in Forming the Set of APEs and Fault Tolerant Capability of the APE Machine

As indicated in Figure 2.1, the program control unit can send out a test signal to every APE channel. Depending on the complexity of the test, the operator could find out from the response how well the APE or APEs of that

channel is functioning before he tries to put that type of APEs into operation. Only a simple existence test for an APE in a specific channel is implemented in our machine: A negative response from this test would indicate either that there is no APE in the set belonging to this type or that none of the APEs of this channel are functioning properly. Then this channel is simply not to be put into operation ...

The freedom to have redundant APEs of the same type operating in parallel as well as to have no APE or no properly functioning APE in a particular channel leads directly to some important features of the APE machine. First, the set of APEs can be grouped together untested and in a random fashion. In the imminent LSI era, an APE being build as a single LSI chip becomes a realistic possibility. The fact that testing is not needed helps to simplify the production of these LSI chips. Ultimately, it is quite conceivable that the APEs could be mass produced at low cost: Whenever a set of APEs is needed to set up an APE machine, they could just be picked up at random from a large stock of untested LSI chips. If too many of them are redundant or perhaps do not function, the operator simply throws more LSI chips into the set of the APEs until he has sufficient number of different types of APEs (i.e. different frequencies) at his disposal for the job. Secondly, the redundant APEs increase the reliability of the APE machine. Of course there are many possible ways a malfunction could occur. All those malfunctions which result in sending shorter output pulses will have no significant effect upon the APE machine if there is at least one APE in each type functioning properly.

2.4 The Structure of an APE

A functional diagram of an APE is given in Figure 2.3. It has a instruction receiver tuned permanently to channel ν_k for the k type APE elements. During the programming mode the transmitter of every APE is being shut off, and the instruction receiver is activated to receive programming instructions from the APE control unit. The program instruction carries the information of how to tune the data receivers A and B as well as what specific operation to perform. This information is stored and decoded by the decoder for function. After completing the programming of all APEs involved in a specific program, the APE machine is switched to the execution mode. The two tunable data receivers A and B are now tuned according to the instruction. The data which they receive are in duty cycle modulation. They are converted into binary numbers and stored by two duty cycle decoders as shown in the figure. These binary numbers x and y are now fed into the stochastic computing element for a specific type of processing according to the value of ξ from the decoder for function . The processing result z is in binary and is again encoded into duty cycle representation before it is transmitted. The clock receiver provides the synchronization for the transmission of the output data, and it also carries the mode control signal. The 'alive' reply signal control is incorporated to answer a test signal from the control unit. The answer is also sent out through the output transmitter. The details of structure and the operation of the APE will be discussed in Chapter 5.

2.5 Power Supply of the APE

The APEs are powered by four solar cells illuminated by incandescent lamps. The details on this part of the project is given in Chpater 7.

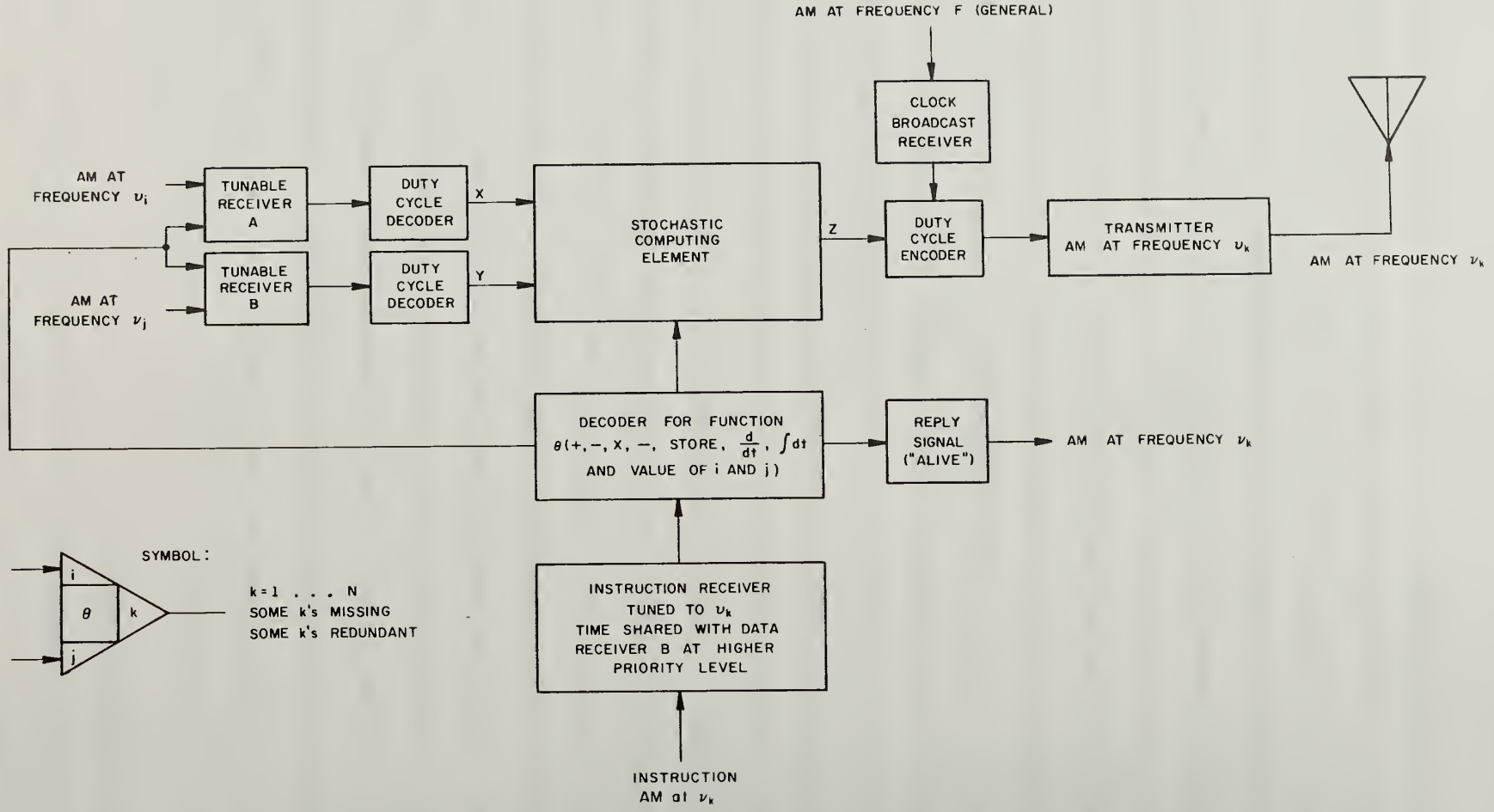


Figure 2.3 The Functional Diagram of the APE

3. STOCHASTIC DATA PROCESSING METHODS OF THE APE MACHINE AND THEIR STOCHASTIC PROPERTIES

3.1 Number Representation in Stochastic Data Processing

During the past decade, considerable interest has arisen in a new way of representing numbers on electronic computers. Unlike the common methods of representing a number by a continuous electrical quantity or by a temporal sequence of discrete electrical quantity, the new method represents a number by the statistical average of a random variable in a discrete stochastic process. Although this is not a very efficient way to represent numbers from the information encoding point of view, extremely simple digital hardware is sufficient to process data in this representation. The most striking advantage of this representation of numbers lies in the fact that any Boolean operation on binary variables corresponds to an arithmetic operation on the numbers corresponding to a sequence of these binary variables. Therefore a simple logic gate can perform a complicated arithmetic operation which would require scores of gates to perform if the numbers were in common representation. Several different types of stochastic number representations[8][10] have been developed. The type being employed for the APE machine is called synchronous random pulse sequence or simple SRPS. In this case, a number is represented by the probability of occurrence of a pulse in any time slot of a clocked time pulse sequence. An example of a SRPS is depicted in Figure 3.1. In terms of mathematical language, a SRPS is a discrete stochastic process denoted by a sequence of statistically independent random variables X_n where X_n is a Boolean variable. For a specific n, it can be either '1' or '0'. The subscript n covers the entire positive integer. The m^{th} order joint distribution function of X_n, is given by $P(X_1, X_2, \ldots X_m)$. Because X_i and X_j with $i \neq j$

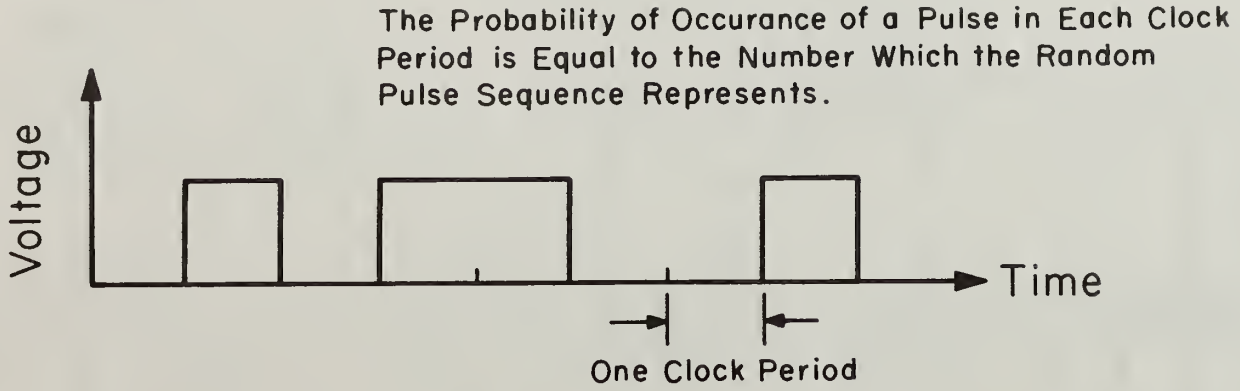


Figure 3.1 A Sample of a Synchronous Random Pulse Sequence

are independent random variables, it follows that

$$P(X_1, X_2, \ldots, X_m) = \prod_{n=1}^{m} P(X_n) \qquad 3.1$$

To map a number into a SRPS, one must first of all normalize the number and express it as a fraction of the full range. Then a SRPS is to be produced such that the probability of occurrence of a pulse in any clock period equals that fraction. In general, the SRPS is a non-stationary and therefore non-ergodic process because the number which it represents is generally a function of time. However, if the time average of a SRPS is computed over a period in which the number it represents is held fixed, as it is precisely the case for the APE, a SRPS can be treated as an ergodic process. It follows that the time average of any sample function of a SRPS is identical to the statistical average. This property allows the conversion of a number from stochastic representation back to the common analog representation simply by integration over a sampling time period.

3.2 Some Simple Boolean Operations on SRPS

Suppose two SRPS's X_n and Y_n represent two numbers x and y respectively. If an AND gate is fed by these two SRPS's, the output is visibly a new SRPS with the probability of having a pulse in any time slot given by $P(X_n=1, Y_n=1)$. Let the resulting SRPS be denoted by Z_n and the number it represents by z. Then the statistical mean of Z_n is given by

$$\text{Exp}\{Z_n\} = P(Z_n=1) = P(X_n=1, Y_n=1) \qquad 3.2$$

If X_n and Y_n are statistically independent, Equation 3.2 becomes

$$\text{Exp}\{Z_n\} = P(X_n=1) \cdot P(Y_n=1) \qquad 3.3$$

Recalling that

$$\text{Exp}\{Z_n\} = P(Z_n=1) = z \qquad 3.4$$

$$\text{Exp}\{X_n\} = P(X_n=1) = x \qquad 3.5$$

$$\text{Exp}\{Y_n\} = P(Y_n=1) = y \qquad 3.6$$

Equation 3.3 becomes

$$z = x \cdot y \qquad 3.7$$

This means that a simple AND gate is what is needed to perform multiplication for numbers in SRPSs. Similarly, it can be shown that two numbers in SRPSs X_n and Y_n are mutually exclusive, their Boolean OR operation corresponds to the addition operation.

3.3 Representation of Negative Numbers by SRPS

Since the range of the value of probability is from zero to one, it becomes necessary to have a suitable transformation before a negative number can be represented by a SRPS. Without loss of generality, one only needs to examine the transformation of a variable in a closed region $\{-1, +1\}$ into another variable in the closed region $\{0, 1\}$. Variable values outside the range of $\{-1, +1\}$ can always be normalized to within this range. To avoid confusion, it is helpful to keep in mind the distinction between three different notations all representing the same quantity but in different representation

X_e: external variable with range $\{-1, +1\}$

x: machine variable with range $\{0, 1\}$

X_n: SRPS representing x

For an APE, the transformation between X_e and x is linear and is given by

$$X_e = 1 - 2x \qquad 3.8$$

With this type of transformation, it can be shown that the SRPS representing x can be easily converted to one representing -x by passing it through an inverter.

3.4 Addition and Subtraction Operations in Terms of Machine Variables

Addition in terms of external variable is given by

$$Z_e = X_e + Y_e \qquad 3.9$$

with X_e and Y_e being the input data and Z_e being the sum. The summation in terms of machine variables can be obtained readily. Let x, y, and z be the machine variables corresponding to Z_e, X_e, and Y_e respectively. According to Equation 3.8

$$\begin{aligned} Z_e &= 1 - 2z \\ X_e &= 1 - 2x \qquad 3.10 \\ Y_e &= 1 - 2y \end{aligned}$$

Substituting Equation 3.10 into Equation 3.9 we get

$$z = x + y - 1/2 \qquad 3.11$$

Similarly for subtraction, given by $Z_e = X_e - Y_e$, the operation in terms of machine variables can be expressed by

$$z = x - y + 1/2 \qquad 3.12$$

The hardware circuit for implementing the addition and subtraction operations is shown in Figure 3.2 The processor itself consists of only two AND gates, one OR gate and an inverter. The variables to be processed, namely

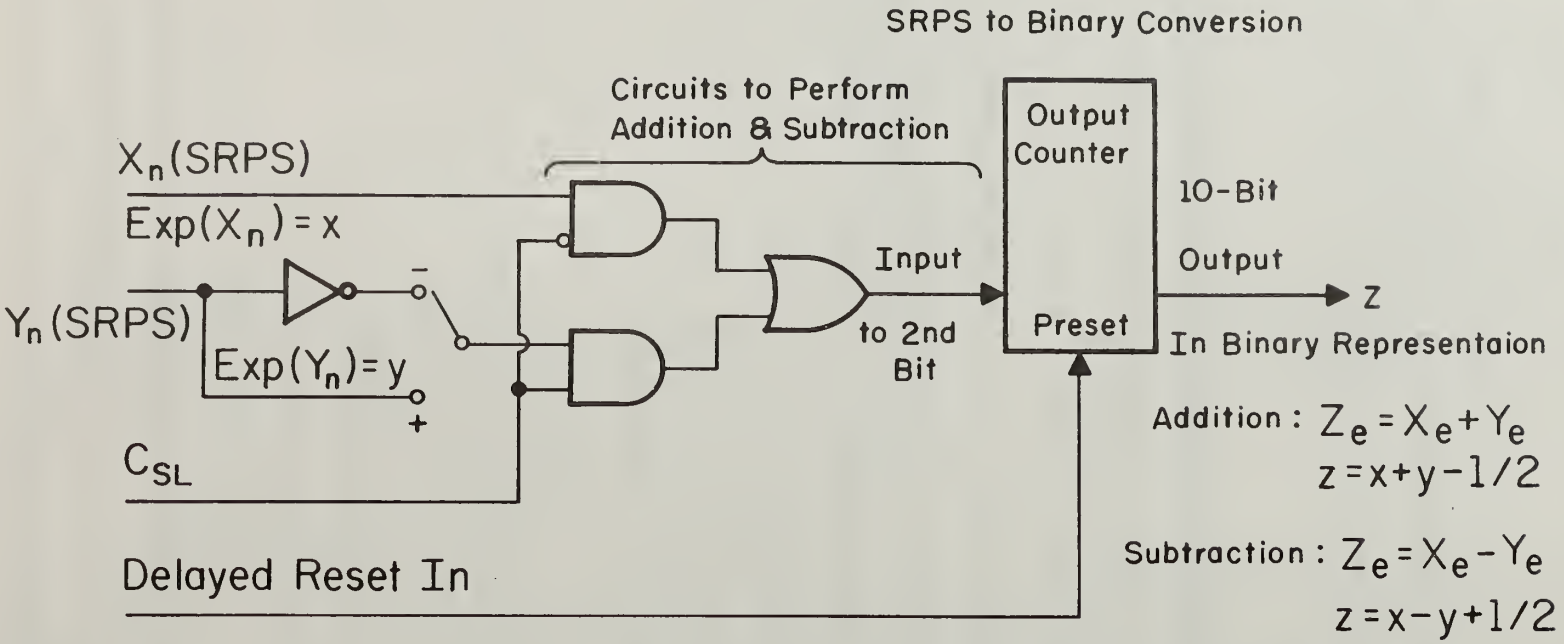


Figure 3.2 Circuits to Perform Addition and Subtraction

X_n and Y_n are in SRPS representation. They are gated alternatively through the AND gates by the complementary gating clock signals. This insures that the two operand SRPSs are mutually exclusive. The gating clock C_{s1} has a duty cycle of precisely 50%. The output from the OR gate would therefore consist of half of each input pulse sequence of X_n and Y_n. This mixed sequence is subsequently gated into the second least significant bit of the output counter. The function of the counter is to convert its SRPS input into the binary number. This is done by counting the number of pulses in the SRPS over as many clock periods as the maximum counts of the counter. The final count would therefore be an estimation of the time average of the input SRPS. In other words, the counter converts the value of a variable in SRPS representation into binary representation. When the input is fed into the 2nd bit of the counter, it amounts to multiplying the input by a factor of two. Because only half of each input SRPSs X_n and Y_n has been gated into the counter, the multiplying factor of two would make the final count correspond to the summation of the two inputs X_n and Y_n. At the beginning of each integrating period, the output counter is preset to half full. Since the counter is operated in a modulo fashion, this means subtracting 1/2 from the result. Therefore the final count of the output counter indicates the estimate of $x + y - 1/2$, which is the summation of the two input variables in terms of machine variables as expressed in Equation 3.11.

As mentioned earlier, inverting a SRPS corresponds to inverting the sign of the number it represents. Therefore, subtraction merely amounts to inverting the subtrahend before doing the addition, as indicated in Figure 3.2.

3.5 Operation of Multiplication in Terms of Machine Variables

As in the case of addition or subtraction, multiplication in terms of machine variable also changes its form under the transformtion of Equation 3.8. In terms of external variables, multiplication is given by

$$Z_e = X_e \cdot Y_e \qquad 3.13$$

where X_e, Y_e and Z_e are the multiplicand, multiplier and the product respectively. Upon transforming these variables into machine variables, the multiplication operation is then given by

$$z = x + y - 2x \cdot y \qquad 3.14$$

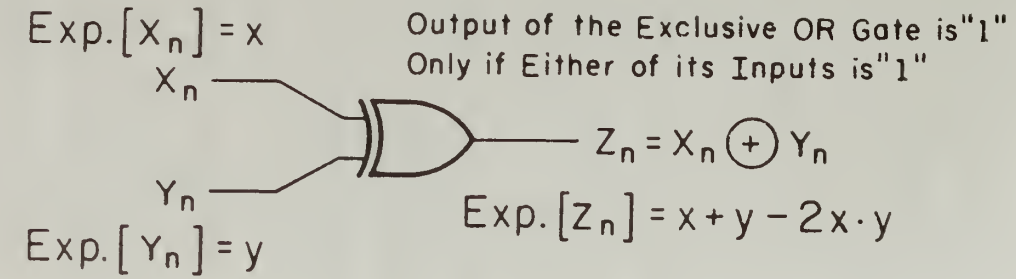
The hardware required to implement Equation 3.14 turns out to be a simple EXCLUSIVE OR gate as shown in Figure 3.3.

3.6 Division in Terms of Machine Variables

Like the other operations discussed above, the division operation in terms of external variables is different from the one in terms of machine variables. Let Z_e, Y_e and X_e be the quotient, numerator and denominator respectively. The relation of their corresponding machine variables can be obtained by applying Equation 3.8 to the expression of division operation. The result is

$$z = -\frac{x - y}{1 - 2x} \qquad 3.15$$

For the implementation of this expression, somewhat more circuitry is involved. To simply the hardware design, Equation 3.15 has to be rewritten to match the hardware characteristics. Since z is necessarily non-negative, z can therefore also be expressed as



$$P(Z_n = 1) = P(X_n = 1, Y_n = 0) + P(X_n = 0\ Y_n = 1)$$
$$= x \cdot (1 - y) + (1 - x)y$$
$$= x + y - 2x \cdot y$$

Figure 3.3 An EXCLUSIVE-OR Gate to Perform Multiplication

$$z = \frac{|x - y|}{|1 - 2x|} \qquad 3.16$$

In absolute value, the numerator and the denominator can be computed more easily. As mentioned in Chapter 2, the machine variables are transmitted and received in duty cycle modulated form, i.e. the width of the transmitted pulse carries the information about the value of the machine variable. Happily enough, a simple EXCLUSIVE OR gate is all it needs to compute the absolute value of the difference of two machine variables in duty cycle modulated form, as depicted in Figure 3.4. To compute the denominator $|1 - 2x|$, the first step is to feed the input decoding counter of the duty cycle decoder (which will be discussed in more detail later on) by the 2nd least significant bit instead of the least significant bit. This corresponds to entering $2 \cdot x$ into the input decoding counter. This counter originally is of 10 bits length to match the range of the numbers it handles. In order to compute the denominator of Equation 3.16, one more bit is added to the most significant position. This extra bit is referred to as the control bit. This control bit is used to control a TRUE/COMPLEMENT gate as shown in Figure 3.5 The remaining 10 bits of the decoding counter represents the data in binary fraction. If x is less than one half, or $2 \cdot x < 1$, the control bit is a '0'. The TRUE/COMPLEMENT gate would gate the complement of the 10-bit binary fraction, i.e., $1 - 2x$, to its output. On the other hand, if $2x \geq 1$ the control bit is '1' and the content of the 10 data bits equals to $2x - 1$. These 10 bits are directly gated to the output of the TRUE/COMPLEMENT gate. Combining these two cases, the output of the TRUE/COMPLEMENT gate is given by $|1 - 2x|$.

After obtaining the numerator $|x - y|$ and the denominator $|1 - 2x|$, it is now necessary to find the quotient. In order to explain the operation

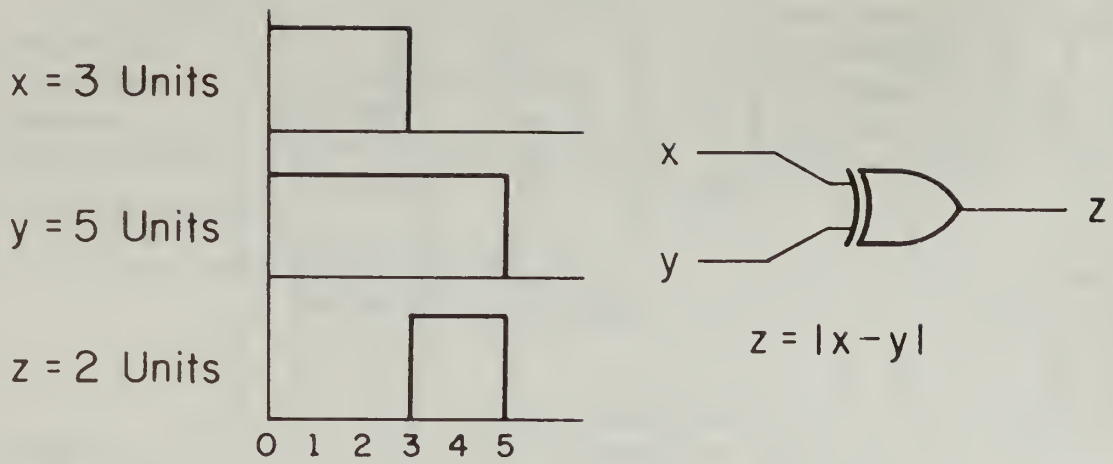


Figure 3.4 An EXCLUSIVE-OR Gate to Compute the Absolute Difference of Two Machine Variable Values

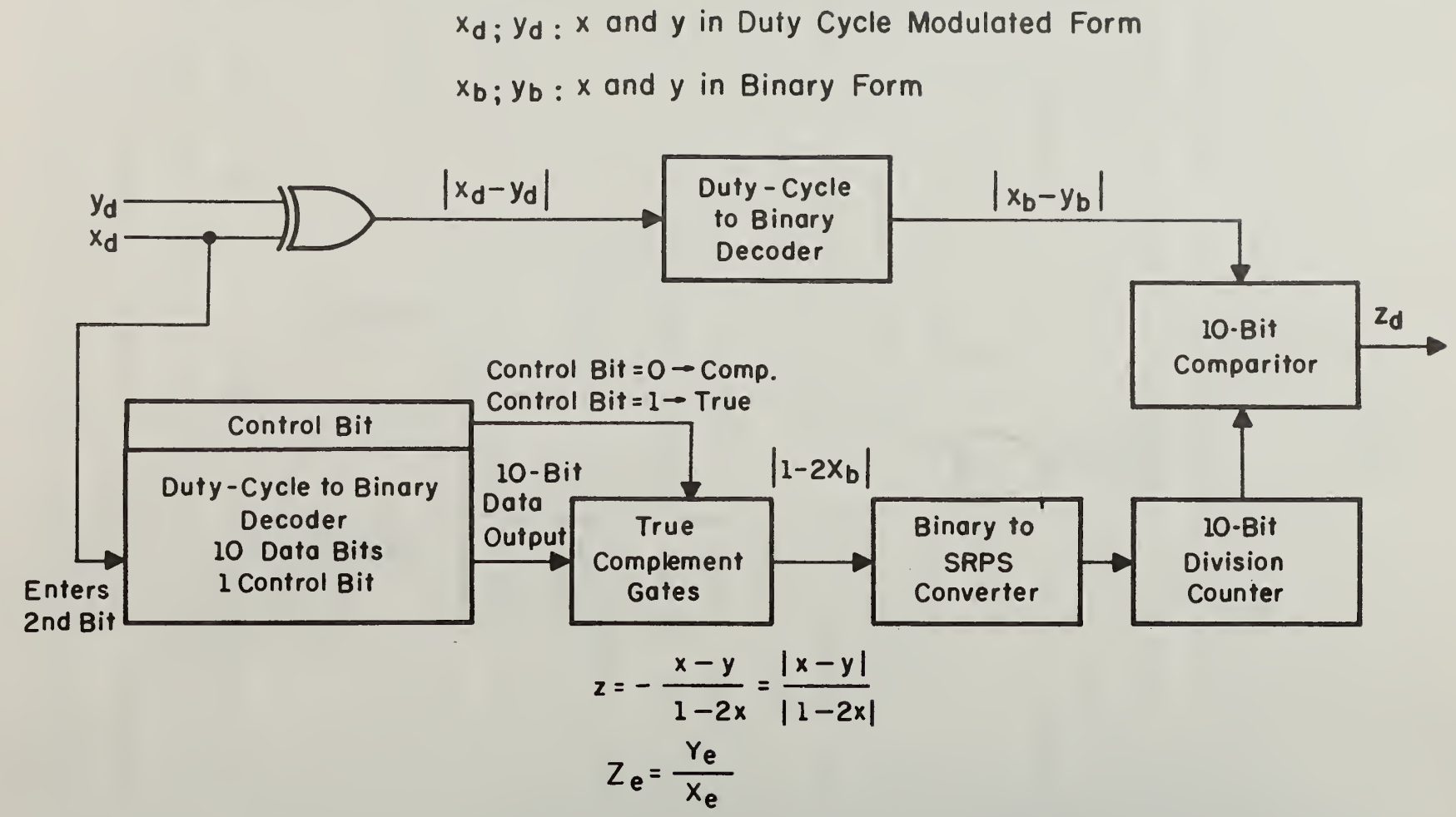


Figure 3.5 The Division Operation in Terms of Machine Variables

clearly, however, it would be best to examine the following case first. Suppose a number a is represented by a SRPS A_n. Let this SRPS A_n be gated into a counter and let the output reading of different bits of the counter represent a binary fraction. Then the time it takes for this counter to accumulate enough counts to make the output reading equal to a given binary fraction n is a random variable with mean value n/a. This relation can be applied to perform the division operation. To do so, let the binary fraction n be equal to $|x - y|$ and let the number a, which the SRPS A_n represents, equal $|1 - 2x|$. Then the quotient $|x - y|/|1 - 2x|$ is obtained by counting the SRPS representing $|1 - 2x|$ until the counter reading the binary fraction is equal to $|x - y|$. The time it takes to accomplish this is then the quotient, already in duty cycle modulation form. The hardware required for the division operation is actually not nearly as complicated as it appears in Figure 3.5. This is because the decoding, encoding and storage circuits are also included in the figure.

3.7 Statistical Properties of Stochastic Processing Operations of the APE

In this section, a statistical analysis is given of 1) the SRPS, 2) the result of arithmetic operations on numbers represented by SRPSs, and 3) the result obtained by cascading several APEs together.

3.7.1 Statistical Properties of an SRPS

As mentioned before, an SRPS corresponds in general to a non-stationary stochastic process. However, it can be treated as ergodic process if the number it represents is held fixed for each computing cycle. Since this is the case for the APE machine, the SRPS is treated as ergodic throughout the remainder of this paper.

3.7.1.1 Statistical Parameters and Sampling Statistics of an SRPS

An SRPS X_n representing a number x has a distribution function $P(X_i=1) = x$ for all i. If an SRPS X_n is given, it is impossible to find out exactly what the number x is in a finite period of time. The best that can be done in a finite period of time is to find an estimate of x. The estimate is obtained by integrating or (for our case) counting over a finite number K of clock periods. The result is called sampling mean and is given by

$$(X_n)_K = \sum_{i=1}^{K} \frac{X_i}{K} \qquad 3.16a$$

This sampling mean is also a random variable and has its mean $\mu_{(X_n)_K}$ and standard deviation $\sigma_{(X_n)_K}$. The objects of the section are to derive these parameters.

Let the probability of having L number of 1's in K number of clock periods for an SRPS X_n be P(K, L). Since the SRPS X_n is a sequence of independent random variables with a distribution function $P(X_i=1) = x$, P(K, L) is binomially distributed and is given by

$$P(K, L) = \frac{K!}{L!(K-L)!} x^{L}(1-x)^{K-L} \qquad 3.17$$

By comparing their definitions, it is not difficult to find that $P(K, L)$ is actually the distribution function for $(X_n)_K$ with $(X_n)_K$ given by L/K. Therefore one can write

$$P\{(X_n)_K\} = \frac{K!}{L!(K-L)!} x^{L}(1-x)^{K-L} \qquad 3.18$$

$$\text{with } (X_n)_K = L/K$$

Since $(X_n)_K$ is binomially distributed, its mean and standard deviation are well known and given by

$$\mu_{(X_n)_K} = x \qquad 3.19$$

$$\sigma_{(X_n)_K} = \sqrt{\frac{x(1-x)}{K}} \qquad 3.20$$

3.7.1.2 The Relation between the Confidence Level, Error, and the Integration Time

The most general approach to this problem uses the Tchebycheff's inequality which states that for a random variable X having an arbitrary distribution with mean μ and standard deviation σ, the following inequality holds for every $\varepsilon > 0$

$$P(|x - \mu| \geq \varepsilon) < \frac{\sigma^2}{\varepsilon^2} \qquad 3.21$$

The confidence level for the random variable X lying within the maximum allowable error ε from its mean μ is given by

$$\alpha = P(|x - \mu| < \varepsilon) \qquad 3.22$$

Because of the fact that

$$P(|x - \mu| < \varepsilon) + P(|x - \mu| > \varepsilon) = 1 \qquad 3.23$$

the confidence level α can also be expressed as

$$\alpha = 1 - P(|x - \mu| \geq \varepsilon) \qquad 3.24$$

Now if the random variable is $(X_n)_K$, Equation 3.21 becomes

$$P(|(X_n)_K - \mu_{(X_n)K}| \geq \varepsilon) \leq \frac{\sigma^2_{(X_n)K}}{\varepsilon^2} \qquad 3.24$$

By combining Equations 3.24, 3.23 and 3.20 one can obtain the relation between confidence level α, maximum allowable error ε, and the minimum number of clock periods K over which the integration must be taken:

$$K \geq \frac{x(1-x)}{\varepsilon^2(1-\alpha)} \qquad 3.25$$

However, recalling that $0 \leq x \leq 1$,

$$x(1 - x) < 1/4 \qquad 3.26$$

By substituting Equation 3.26 into Equation 3.25, one gets

$$K \geq \frac{1}{4\varepsilon^2(1-\alpha)} \qquad 3.27$$

For example, in order to find out the time average of a SRPS by taking the average over 5×10^4 clock periods, one will have at least 95% confidence in the reading being within 1% of the exactly correct result. The fact that at least 95% confidence instead of simple 95% confidence is due to the 'less than or equal to' relation in the Tchebycheff's inequality.

Equation 3.27 is derived from Tchebycheff's inequality, which is applicable to random variable with any distribution function. Consequently, Equation 3.27 represents a requirement somewhat unnecessarily restrictive for the binomially distributed $(X_n)_K$.

A less general approach, which leads to a weaker requirement than that of Equation 3.27, takes into account the specific distribution of $(X_n)_K$. Based on central limit theorem of statistics, this approach approximates the binomial distrubition of $(X_n)_K$ with a normal distribution.

$$P\{(X_n)_K\} = \frac{1}{\sqrt{2\pi}\,\sigma_{(X_n)_K}} e^{-\frac{[(X_n)_K - x]^2}{2\sigma^2_{(X_n)_K}}} \qquad 3.28$$

The confidence level α with maximum allowable error ε is given by

$$\alpha = \int_{x-\varepsilon}^{x+\varepsilon} P\{(X_n)_K\}\, d(X_n)_K \qquad 3.29$$

By defining a new variable

$$Z = \frac{\{(X_n)_K - x\}}{\sqrt{2}\,\sigma_{(X_n)_K}} \qquad 3.30$$

and substituting Equation 3.28 into 3.29, it becomes

$$\alpha = \frac{1}{\sqrt{\pi}} \int_{-w}^{w} e^{-Z^2}\, dZ \qquad 3.31$$

where $W = \frac{\varepsilon}{\sqrt{2}\,\sigma_{(X_n)_K}}$ 3.32

In terms of the error function ERF, Equation 3.31 becomes

$$\alpha = ERF\left\{\frac{\varepsilon}{\sqrt{2}\,\sigma_{(X_n)_K}}\right\} \qquad 3.33$$

substituting Equations 3.20, 3.26 into Equation 3.33, one can write

$$\alpha = ERF\{\varepsilon\sqrt{2K}\} \qquad 3.34$$

This gives the relation between α, ε, and K. The validity of the approximation being made in this approach requires a large K. In the case of the APE machine K is over 10^5, Equation 3.34 gives a good approximation* to the relation between the parameters involved. For the example of 95% confidence with 1% maximum allowable error, the integration must be taken over approximately $K = 10^4$ periods. Comparing with the result $K = 5 \times 10^4$ obtained from

* According to the discussion given in Reference (20), the error produced by the approximation for the case in which $K = 10^5$ and $Exp[(X_n)_K] = \frac{1}{2}$ is less

the previous approach, they differ by a factor of 5. However, they are still consistent because what the previous approcah concludes is that for $K = 5 \times 10^4$, the reading provides at least 95% confidence and accurates to within 1%. The result of the second approach obviously agrees with it.

3.7.2 Statistical Properties of the Result of Addition and Subtraction by an APE

A close examination of the operation of addition reveals that integrating the sum over K clock periods corresponds to integrating the addend and the augend over $\frac{K}{2}$ clock periods each. Let x, y be the addend and the augend respectively. The sum in terms of machine number is given by $z = x + y - \frac{1}{2}$ as shown before. According to Equations 3.19 and 3.20 the sampling statistics of a sample size of $H = \frac{K}{2}$ from X_n and Y_n are given by

$$\mu_{(X_n)_H} = \text{Exp}\{\sum_{i=1}^{H} \frac{X_i}{H}\} = x \qquad 3.35$$

$$\sigma_{(X_n)_H} = \frac{\sqrt{x(1-X)}}{H} \qquad 3.36$$

$$\mu_{(Y_n)_H} = \text{Exp}\{\sum_{i=1}^{H} \frac{Y_i}{H}\} = y \qquad 3.37$$

$$\sigma_{(Y_n)_H} = \frac{\sqrt{y(1-y)}}{H} \qquad 3.38$$

It can be shown that the sampling mean of the sum is given by

$$\mu_{(Z_n)_K} = \mu_{(X_n)_H} + \mu_{(X_n)_H} - \frac{1}{2} \qquad 3.39$$

By substituting Equations 3.35, 3.37 into Equation 3.39, one can write

$$\mu_{(Z_n)_K} = x + y - \frac{1}{2} \qquad 3.40$$

It can also be shown that the sampling standard deviation of the sum is given by

$$\sigma_{(Z_n)_K} = \sqrt{\frac{x(1-x)}{2K} + \frac{y(1-y)}{2K}} \qquad 3.41$$

Substituting Equation 3.41 into Tchebycheff's inequality, one gets

$$(1-\alpha) < \frac{x(1-x) + y(1-y)}{2K\varepsilon^2} \qquad 3.42$$

Recall that $x(1-x) \leq \frac{1}{4}$, $y(1-y) \leq \frac{1}{4}$. Consequently, Equation 3.42 can be written as

$$K \geq \frac{1}{4\varepsilon^2(1-\alpha)} \qquad 3.43$$

This gives a relation between confidence level α, maximum allowable error ε and the sampling size K, based on the 1st approach. The result based on the 2nd approach can also be obtained by replacing the $\sigma_{(X_n)_K}$ in Equation 3.33 with $\sigma_{(Z_n)_K}$. The result is given by

$$\alpha = \mathrm{ERF}\{\varepsilon\sqrt{2K}\} \qquad 3.44$$

The subtraction operation has the same statistical properties as the addition operation, because subtraction is done exactly the same way as the addition operation, except that the subtrahand SRPS is inverted before it is added to the minuend. This only amounts to changing the value x of the subtrahand SRPS X_n to (1-x) before the addition operation takes place.

3.7.3 Statistical Properties of the Product of Multiplication Operation

Let z, x and y represent the product, the multiplicand and the multiplier. As mentioned in Section 3.5, the sequence obtained at the output of the EXCLUSIVE-OR gate is an SRPS representing the product with the mean value given by

$$\mu_Z = z$$

$$= x + y - 2x\cdot y \qquad 3.45$$

All the result about the statistical properties of an SRPS derived in 3.7.1 are therefore applicable for the product. In particular, the fluctuation of the samples is given by

$$\sigma(Z_n)_K = \sqrt{\frac{(x+y - 2x\cdot y)(1-x -y + 2x\cdot y)}{K}} \qquad 3.46$$

The relations between α, K, and ε for the product is identical to the ones given in Equation 3.27 and Equation 3.34.

3.7.4 Statistical Properties of the Quotient in Division Operations

It was shown in Section 3.6 that the Quotient Z_d of the division operation is a random variable. Its statistical properties is now being investigated. The distribution function of Z_d corresponds to the probabilities of getting a fixed number of pulses for different numbers of clock periods. Let T_i be the number of clock periods between the i^{th} pulse and its following pulse in a sample function of an SRPS. The total number of clock periods T_n for an SRPS D_n, representing the denominator in Equation 3.16 in the case of division operation, to produce N number of pulses is therefore given by

$$T_n = \sum_{i=1}^{N} T_i \qquad N < T_n < \infty \qquad 3.47$$

Theoretically, T_n ranges from N and infinity. For a real computer, integers cannot be larger than a limit determined by the particular structure of the computer. For the APE machine, this limit is $K = 2^{17}$. Therefore T_n actually ranges from N to K. The normalized value of T_n, given by $t_n = \frac{T_n}{K}$, corresponds to the final result of the division operation in the form of a binary fraction. To examine the statistical properties of t_n, it is better to start from the distribution function of T_i. Since T_i is the number of clock periods between two consecutive pulses in the SRPS D_n representing the number d, the probability that $T_i = n$ is given by the

probability of having no pulse in (n-1) clock periods times the probability of having a pulse in the n^{th} period. One can, therefore, write

$$P(T_i=n) = d(1-d)^{n-1} \qquad 3.48$$

Let b denote (1-d) and D be the differential operator $\frac{d}{db}$. The expectation value of T_i can then be expressed as

$$\begin{aligned} Exp\{T_i\} &= \sum_{n=1}^{\infty} nd\cdot b^{n-1} \\ &= d \sum_{n=1}^{\infty} D\cdot b^{n} \\ &= d\cdot D \sum_{n=1}^{\infty} b^{n} \\ &= d\cdot D\left(\frac{b}{1-b}\right) \qquad 3.49 \\ &= \frac{1}{d} \end{aligned}$$

The mean value of T_i^2 is given by

$$\begin{aligned} Exp\{T_i^2\} &= \sum_{n=1}^{\infty} n^2\cdot d\cdot b^{n-1} \\ &= d \sum_{n=1}^{\infty} (n+1)n\cdot b^{n-1} - d \sum_{n=1}^{\infty} n\cdot b^{n-1} \qquad 3.50 \end{aligned}$$

Upon substituting Equation 3.49 into the 2nd term of Equation 3.50, one gets

$$\begin{aligned} Exp\{T_i^2\} &= d \sum_{n=1}^{\infty} D^2\cdot b^{n+1} - \frac{1}{d} \\ &= d\cdot D^2\left(\frac{b^2}{1-b}\right) - \frac{1}{d} \\ &= \frac{2}{d^2} - \frac{1}{d} \qquad 3.50 \end{aligned}$$

The standard deviation σ_T is given by

$$\sigma_T = \sqrt{Var\{T_i\}}$$

$$= \sqrt{Exp\{T_i^2\} - Exp^2\{T_i\}} \qquad 3.50a$$

By substituting Equations 3.49 and 3.50 into Equation 3.50a, one gets

$$\sigma_T = \sqrt{\frac{2}{d^2} - \frac{1}{d} - \frac{1}{d^2}}$$

$$= \sqrt{\frac{1}{d^2} - \frac{1}{d}} \qquad 3.51$$

The mean and standard deviation of t_N can now be obtained as follows

$$Exp\{t_N\} = \frac{1}{K} \sum_{T_1=1}^{\infty} \sum_{T_2=1}^{\infty} \cdots \sum_{T_N=1}^{\infty} \sum_{i=1}^{N} T_i \cdot P(T_1, T_2, \ldots, T_N) \qquad 3.52$$

Since T_i and T_j, $i \neq j$, are independent random variables

$$P(T_1, T_2, \ldots, T_N) = P(T_1)P(T_2) \ldots P(T_N) \qquad 3.53$$

Together with the fact that

$$\sum_{T_i=1}^{\infty} P(T_i) = 1 \quad \text{for all } i \qquad 3.54$$

Equation 3.52 becomes

$$Exp\{t_N\} = \frac{1}{K} \sum_{i=1}^{N} Exp\{T_i\} = N \cdot \frac{d}{K} \qquad 3.55$$

Now the variance of t_N can be expressed in

$$Var\{t_N\} = Exp[(t_N - Exp\{t_N\})^2] \qquad 3.56$$

Combining Equations 3.55 and 3.56 one obtains

$$\mathrm{Var}\{t_N\} = \mathrm{Exp}\left[\frac{1}{K^2}\left(\sum_{i=1}^{N} T_i - \sum_{i=1}^{N} \mathrm{Exp}\{T_i\}\right)^2\right]$$

$$= \frac{1}{K^2}\,\mathrm{Exp}\left\{\left[\sum_{i=1}^{N}\left(T_i - \mathrm{Exp}\{T_i\}\right)\right]^2\right\}$$

$$= \frac{1}{K^2}\,\mathrm{Exp}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[T_i - \mathrm{Exp}\{T_i\}\right]\left[T_j - \mathrm{Exp}\{T_j\}\right] \qquad 3.57$$

Because of the fact that T_i and T_j, for $i \neq j$, are statistical independent and that $\sum_{T_i=1}^{\infty} P(T_i) = 1$, the expectation operator and the summation operators in Equation 3.57 can be shown to be commutative, i.e.,

$$\mathrm{Var}\{t_N\} = \frac{1}{K^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{Exp}\{[T_i - \mathrm{Exp}(T_i)][T_j - \mathrm{Exp}(T_j)]\} \qquad 3.58$$

Again because T_i and T_j, for $i \neq j$, are independent, all terms under the summation sign with $i \neq j$ in Equation 3.58 are zero. There Equation 3.58 becomes

$$\mathrm{Var}\{t_N\} = \frac{1}{K^2}\sum_{i=1}^{N}\mathrm{Exp}\{(T_i - \mathrm{Exp}(T_i)^2)\}$$

$$= \frac{1}{K^2}\sum_{i=1}^{N}\mathrm{Var}\{T_i\} \qquad 3.59$$

Because all the T_i's have identical distribution, they therefore have identical variance. Consequently Equation 3.59 becomes

$$\mathrm{Var}\{t_N\} = \frac{N}{K^2}\mathrm{Var}\{T_i\} \qquad 3.60$$

It follows that the standard deviation is given by

$$\sigma_{t_N} = \frac{\sqrt{N}\sigma_T}{K} \qquad 3.61$$

Combining Equations 3.51 and 3.61, one gets

$$\sigma_{t_N} = \sqrt{\left(\frac{1}{K}\right)\left(\frac{N}{K}\right)\left(\frac{1}{d^2} - \frac{1}{d}\right)} \qquad 3.62$$

Recall that the division operation is performed by counting the pulses of an SRPS D_n representing the divisor $|1-2x|$, shown in Equation 3.16, over a period such that the final reading of the counter in binary fraction equals to the dividend $|x-y|$. The term $(\frac{N}{K})$ in Equation 3.62 is the normalized value of the number of pulses being counted. Therefore the term $\frac{N}{K}$ actually corresponds the dividend $|x-y|$ in the division operation. The parameter d is the divisor in machine number. It has the value of $|1-2x|$. After replacing d, and $\frac{N}{K}$ with the expression in terms of x and y, Equation 3.62 becomes

$$\sigma_{t_N} = \sqrt{\frac{|x-y|}{K}\left(\frac{1}{|1-2x|^2} - \frac{1}{|1-2x|}\right)} \qquad 3.63$$

The relation between α, ε and K according to the Tchebycheff's inequality can be obtained by combining Equations 3.63 and 3.27. The result is given by

$$K \geq \frac{|x-y|}{(1-\alpha)\varepsilon^2}\left(\frac{1}{|1-2x|^2} - \frac{1}{|1-2x|}\right) \qquad 3.64$$

Based on the central limit theorem, the relation between α, ε and K can be approximated by

$$\alpha = \mathrm{ERF}\left[\frac{\varepsilon}{\sqrt{2\frac{|x-y|}{K}\left(\frac{1}{|x-y|^2} - \frac{1}{|x-y|}\right)}}\right] \qquad 3.65$$

3.7.5 Statistical Properties of the Result of Cascading Two Operations

The statistical properties of the basic arithmetical operations have been examined. They are investigated under the condition in which the inputs

to the stochastic processor are deterministic. Now the problem of probabilistic inputs is to be investigated. This is the case when two levels of stochastic processors are operated in cascade, like the example shown in Figure 2.2. The stochastic processors in the second level are driven by the outputs of the first level ones. These APE outputs are the sampling means, instead of the means, of the output SRPSs of the first level APEs and are therefore random variables. Let the outputs of two first level APEs be denoted by X and Y. They are random variables with normal distribution, according to central limit theorem, as discussed above. When X and Y are fed into another APE for further processing, the result can be represented in general by

$$Z = f(X, Y) \qquad 3.66$$

Theoretically, the statistical properties of Z can be derived from those of X and Y. Unfortunately, even for some rather simple function f(X, Y), to obtain the result in explicit closed form becomes very much involved. A direct approach to this problem is to find out the distribution of Z from those of X and Y and from f(X, Y). However, the following approach dodges the evaluation of the distribution of Z and simplifies the problem somewhat. To compute the expectation value of Z, one writes

$$\overline{Z} = \int_0^1 \int_0^1 f(X, Y)P(X, Y)\, dX \cdot dY \qquad 3.67$$

where P(X, Y) is the joint distribution function of the input variables. For the standard deviation σ_Z one can first find out $\overline{Z^2}$, the simplified notation for $Exp[Z^2]$, by

$$\overline{Z^2} = \int_0^1 \int_0^1 f^2(X, Y)P(X, Y)dX \cdot dY \qquad 3.68$$

and makes use of the expression $\sigma_Z^2 = \overline{Z^2} - \overline{Z}^2$ 3.69

The standard deviation σ_Z is then given by

$$\sigma_Z = \sqrt{\int_0^1\int_0^1 f^2(X,Y)P(X,Y)dX\cdot dY\} - \{\int_0^1\int_0^1 f(X,Y)P(X,Y)dX\cdot dY\}^2} \qquad 3.70$$

This equation indicates the fluctuation of the result due to that of the inputs. For example, if the operation is multiplication with the multiplicand and the multiplier given by the sampling means of two SRPSs and with the product in the form of another SRPS, the mean value of the product SRPS, instead of being a deterministic number representing the exact value of the product, is a random variable and fluctuates with a standard deviation given by Equation 3.70. The relation between the confidence level, maximum allowable error, and standard deviation as given in 3.33 is apparently applicable here. Let the confidence level, error and standard deviation associated with the fluctuation of the mean value of the output random variable due to the input fluctuation be α_Z, ε_Z and σ_Z respectively. Then the relation between them is given by

$$\alpha_Z = \mathrm{ERF}\{\frac{\varepsilon_Z}{\sqrt{2}\,\sigma_Z}\} \qquad 3.71$$

Let the confidence level, error and standard deviation associated with the estimation, with a finite sample size of K samples, of the output variable having deterministic mean value be α_K, ε_K and σ_K. Their relation has been derived before and is given by

$$\alpha_K = \mathrm{ERF}\{\frac{\varepsilon_K}{\sqrt{2}\,\sigma_K}\} \qquad 3.72$$

Now, let the confidence level and error due to the combination of the two cases mentioned before be α_T, ε_T. It is not difficult to see that

$$\alpha_T = \alpha_Z \cdot \alpha_K$$

$$\varepsilon_T = \varepsilon_Z + \varepsilon_K \qquad 3.73$$

Combining Equation 3.71, 3.72 and 3.73 one gets

$$\alpha_T = \mathrm{ERF}\{\frac{\varepsilon_Z}{\sqrt{2}\,\sigma_Z}\}\{\mathrm{ERF}\,\frac{\varepsilon_K}{\sqrt{2}\,\sigma_K}\} \qquad 3.74$$

As an example, let us consider an addition operation with the addend and augend coming from the outputs of other APEs. According to the previous notation, the addend and augend are denoted by $(X_n)_K$ and $(Y_n)_K$ to indicate that they are obtained from SRPSs X_n and Y_n by averaging over K clock periods. In this section, the subscripts are dropped to simplify the notation so that $(X_n)_K$ is now denoted by X and $(Y_n)_K$, by Y. The addition operation is therefore given by

$$f(X,Y) = Z = X + Y - \frac{1}{2} \qquad 3.75$$

To find σ_Z according to Equation 3.70, one must first find out P(X,Y). Since X and Y are independent random variables, one can write

$$P(X,Y) = P(X)\cdot P(Y) \qquad 3.76$$

By substituting Equation 3.28 into 3.76, one gets

$$P(X,Y) = \frac{1}{2\pi\sigma_X\sigma_Y}\, e^{-\left[\frac{(X-\mu_X)^2}{2\sigma_X^2} + \frac{(Y-\mu_Y)^2}{2\sigma_Y^2}\right]} \qquad 3.77$$

where μ_X and μ_Y are the mean values of X and Y respectively and σ_X and σ_Y are the standard deviations of X and Y respectively. Upon substituting equations 3.75 and 3.77 into Equation 3.67, one gets

$$\bar{Z} = \int_0^1 \int_0^1 (X + Y - \tfrac{1}{2}) \cdot \frac{1}{2\pi\sigma_X\sigma_X} \, e^{-\left[\frac{(X-\mu_X)^2}{2\sigma_X^2} + \frac{(Y-\mu_Y)^2}{2\sigma_Y^2}\right]} \, dXdY$$

$$= \mu_X + \mu_Y - \frac{1}{2} \qquad 3.78$$

Similarly, one can compute the expectation value of Z^2, by

$$\bar{Z}^2 = \int_0^1 \int_0^1 (X + Y - \tfrac{1}{2})^2 \cdot \frac{1}{2\pi\sigma_X\sigma_Y} \, e^{-\left[\frac{(X-\mu_X)^2}{2\sigma_X^2} + \frac{(Y-\mu_Y)^2}{2\sigma_Y^2}\right]} \, dXdY$$

$$= \sigma_X^2 + \mu_X^2 + \sigma_Y^2 + \mu_Y^2 + \frac{1}{4} + 2\mu_X\mu_Y - \mu_X - \mu_Y \qquad 3.79$$

The standard deviation σ_Z is therefore

$$\sigma_Z = \sqrt{\bar{Z}^2 - \bar{Z}^2} \qquad 3.79A$$

Upon substituting Equations 3.78 and 3.79 into Equation 3.79A one finds that many terms in the righthand side of Equation 3.79A cancel out, so that

$$\sigma_Z = \sqrt{\sigma_X^2 + \sigma_Y^2} \qquad 3.80$$

Combining Equations 3.80 into 3.74 one arrives at

$$\alpha_T = \mathrm{ERF} \, \frac{\varepsilon_Z}{\sqrt{2}\sqrt{\sigma_X^2 + \sigma_Y^2}} \, \mathrm{ERF} \, \frac{\varepsilon_K}{\sqrt{2}\,\sigma_K} \qquad 3.81$$

Recall that α_X represents the fluctuation of the estimation of the mean value x of the random variable X by averaging over K samples. According to Equation 3.20

$$\sigma_X = \sqrt{\frac{x(1 - x)}{K}} \qquad 3.82$$

However, $x(1 - x)$ is always less than $\frac{1}{4}$. Therefore, one can write

$$\sigma_X^2 \leq \frac{1}{4K} \qquad 3.83$$

Similarly

$$\sigma_Y^2 \leq \frac{1}{4K} \qquad 3.84$$

$$\sigma_K^2 \leq \frac{1}{4K} \qquad 3.85$$

Upon substituting these expressions into Equation 3.81, it becomes,

$$\alpha_T \geq \mathrm{ERF}[\varepsilon_Z\sqrt{K}] \cdot \mathrm{ERF}[\varepsilon_K\sqrt{2K}] \qquad 3.86$$

As an example, let $K = 13 \times 10^4$, $\varepsilon_Z = 0.5\%$, $\varepsilon_K = 0.5\%$ so that $\varepsilon_T = 1\%$, one gets

$$\alpha_T \geq \mathrm{ERF}\{1.80\}\mathrm{ERF}\{2.55\} \qquad 3.87$$

$$\geq 0.988$$

In other words, if integration is taken over $K = 13 \times 10^4$ number of clock periods, the output reading of this cascaded operation would be within 1% of its correct result with 98.8% confidence.

3.7.6 Statistical Properties of the Results of Differential and Integral Operations

The differential operation is actually a subtraction operation. Therefore all the statistical properties derived for subtraction operation are applicable. The statistical properties of the result of the integral operation are similar to those of the addition operation with nondeterministic inputs. Therefore, the results derived in Section 3.7.5 must be applied to compute the relation between confidence level, accuracy and sampling time.

4. THE GENERATION OF THE SYNCHRONOUS RANDOM PULSE SEQUENCES

4.1 General Requirements on the SRPSs Used in the APEs

There are four basic requirements on the SRPSs used in the APEs: a) It has been shown in the previous chapters that numbers represented by SRPSs can be operated upon with great ease. However, the significance of this advantage depends on the ease with which a number can be converted to and from an SRPS. Therefore, the conversion must be able to be accomplished with simple circuits. b) Most of the operations of the APEs involve two operands. It is therefore required to have two SRPSs representing two different operands. These two SRPSs must be statistically independent as required in the multiplication operation. c) The APEs obtain the power remotely. The maximum power available to an APE is less than 100 mW. Therefore, the converter must consume very little power. It is this fact that precludes the well-tested method of converting a number into SRPS by means of noise diodes and thresholding technique. d) Because the number to be converted might be a function of time, the resulting SRPS must be able to follow this change and reach a new steady state within a small fraction of a computing cycle.

4.2 Conversion of a Number into a SRPS

The basic idea of converting a number into its SRPS representation is to compare the number with the output of a noise source. There are many approachs to the conversion based on different ways to make comparisons and different principles to generate noise[12][13][14][15]. The one most suitable for the APE machine is shown in Figure 4.1. The number to be converted is

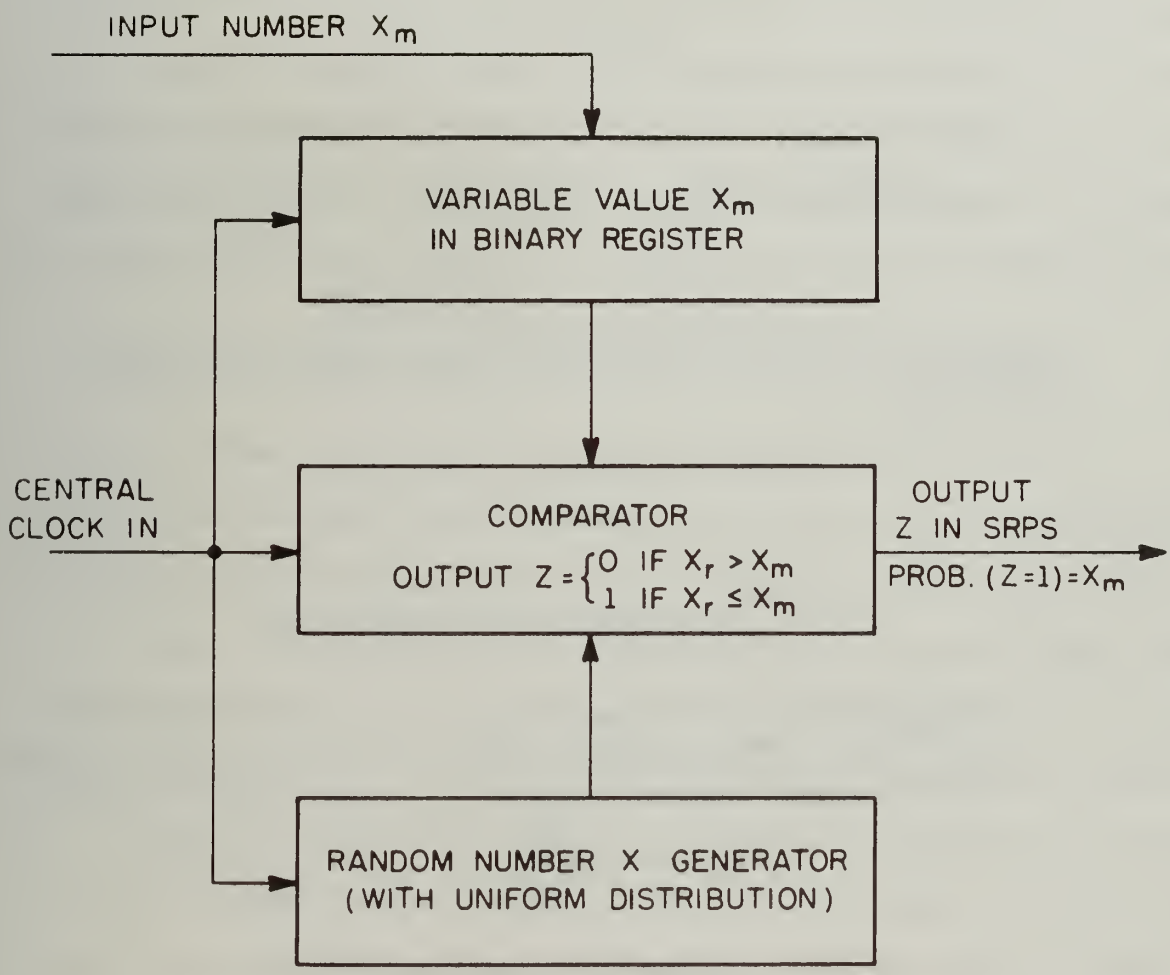


Figure 4.1 Digital to SRPS Converter

in binary representation and denoted by X_m. This number is stored in a binary register. The length of the register is 10 bits for the APE machine. This 10-bit binary number is compared with the output of a 10-bit random number generator continuously. The random number generator produces a new 10-bit binary number at the beginning of every clock period. The 10-bit random binary number has a uniform distribution function. The comparator output is '1' if the 10-bit binary random number is smaller or equal to the 10-bit binary number X_m. It is a '0' otherwise. Because the random number X_r is uniformly distributed, the output of the comparator is a SRPS with the probability of occurrence of a '1' in each clock period equal to X_m.

4.3 Generation of a 10-bit Pseudorandom Binary Number with Uniform Distribution

It is rather involved to generate a truly random number with predetermined statistical properties. However, certain types of well-defined periodic signals could appear to be random if the periods of the signals are much longer than the observation period. One example of this type of signal is the pseudorandom binary sequence. It can be generated with an extremely long period and with very simple hardware. The numbers are generated by modulo-2 linear recurrence technique long used to generate binary code for communications. In this method, a shift register of n stages is connected with a feedback path as shown in Figure 4.2. If the function $f(x_1, x_2, \ldots, x_n)$ can be expressed as a linear combination of the output of various stages of the register, this type of circuit is referred to as linear feedback register. If the register is initially in a state where $x_1 = a_1$, $x_2 = a_2$, $\ldots$, $x_n = a_n$, the next state of register is completely determined and can be expressed by

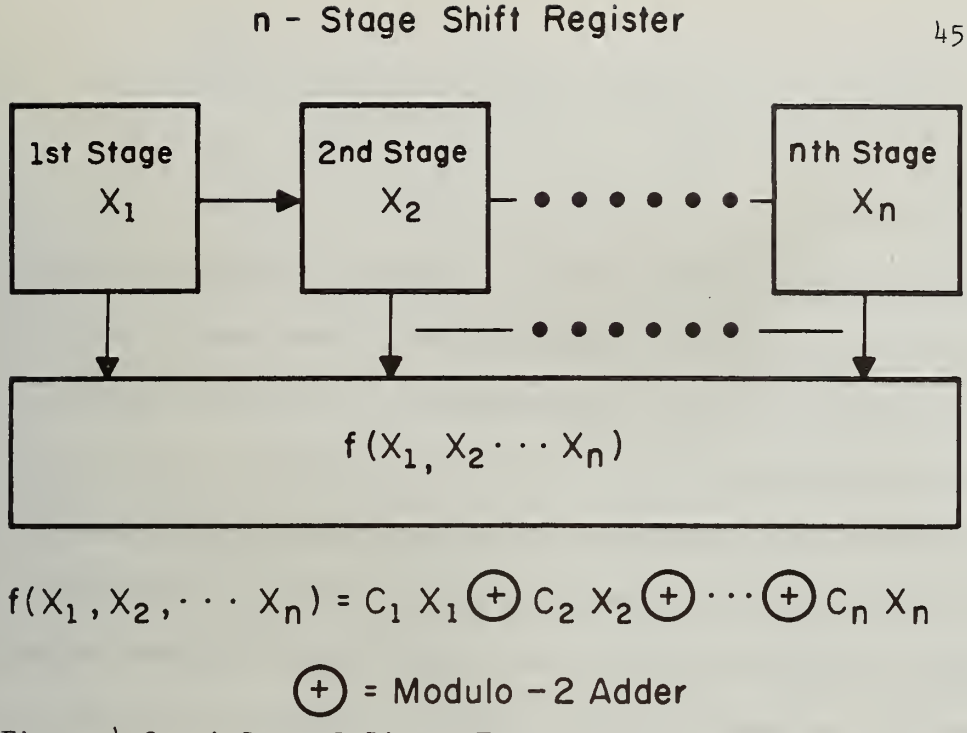


Figure 4.2 A General Linear Feedback Shift Register Arrangement

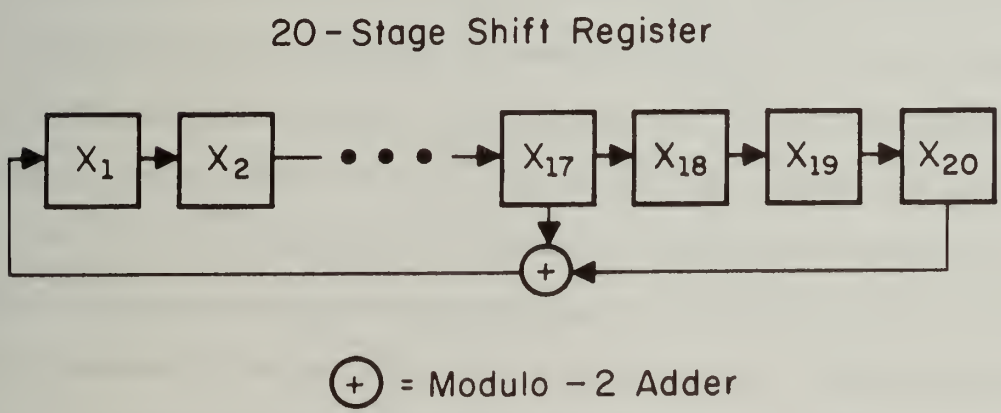


Figure 4.3 The Arrangement for the Generation of a 20-Bit Maximum Length Sequence

$$x_1' = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$$
$$x_i = x_i - 1 \qquad i = 2, 3, \ldots, n \qquad 4.1$$

where x_i' denotes the current value of the i^{th} stage, x_i denotes the value of the i^{th} stage during the last clock period. This means that a new n-bit binary number, related to the last one by Equation 4.1, is generated by the register every clock period.

It is obvious that a different set of coefficient c_i corresponds to a different sequence of n-bit binary number produced by the register. With an appropriate set of coefficients, the length of the sequence before it repeats itself can be made to be $2^n - 1$, which is only one less than the maximum number of states a n-stage register can possibly be in. For a 20 stages feedback shift register, this corresponds to a sequence of more than one million binary numbers before any periodicity occurs. It can be shown that this maximal-length sequence with a period of $2^n - 1$ can be obtained if the set of coefficient c_i's equal to the coefficients of a primitive n^{th} degree polynomial[16]. For a 20 stages feedback register, the set of coefficient has been worked out in reference (17). The result is that all coefficients equal to zero except $c_{17} = c_{20} = 1$, as depicted in Figure 4.3. The first ten bits of the output of the register are used to form 10-bit binary number while the remaining ten bits are used to form another 10-bit binary number. If these two binary numbers are taken once after every shifting of the register to form two binary number sequences, they can be shown, by following a similar argument as presented in reference (14), to be two statistically independent pseudo-random binary number sequences with uniform distribution. They are therefore being used as the noise sources for the APE machine.

4.4 The Singular State of a Linear Feedback Shift Register

A n-stage feedback shift register has 2^n different states. A maximal length sequence will go through every state except the all-zero state. A close look at Figure 4.3 reveals that once a linear feedback shift register is in this all-zero state it remains there indefinitely. If this is the case, the feedback shift register ceases to generate pseudorandom numbers. Therefore, protective circuitry must be used to provide continuous monitoring of the feedback shift register. If it ever gets into this all-zero singular state for some reason, the protective circuit must be able to pull it out. A simple circuit to do this is outlined in Figure 4.4. It employs a 5-stage counter driven by the same clock that drives the shift register. The reset of the counter is connected to the output of any stage while the carry output of the counter is added, modulo-2, to the output of an arbitrary stage. The sum is then fed into the subsequent stage. If the X_i's are not all zero, the feedback shift register will not be in the all-zero state under proper operating conditions. In this case, the reset line will be in logical '1' level at least once every twenty clock periods. As a consequence, the 5-bit counter will be reset before the carry bit becomes '1'. Hence, the carry output is constantly '0'. The output of the modulo-2 adder is therefore identical to the output of the stage denoted by X_ℓ. In other words, the protective circuit will have no effect on the operation in this case. However, if the feedback shift register slips into the all-zero state, the carry-out will be at logical '1' level after 31 shifts because of the lack of the reset signal. A logical '1' signal in the carry-out line when the feedback shift register is in the all-zero state results in inserting a '1' into $X_\ell + 1$ stage during the next clock period. Therefore, the feedback shift register is again put back into a non-singular state.

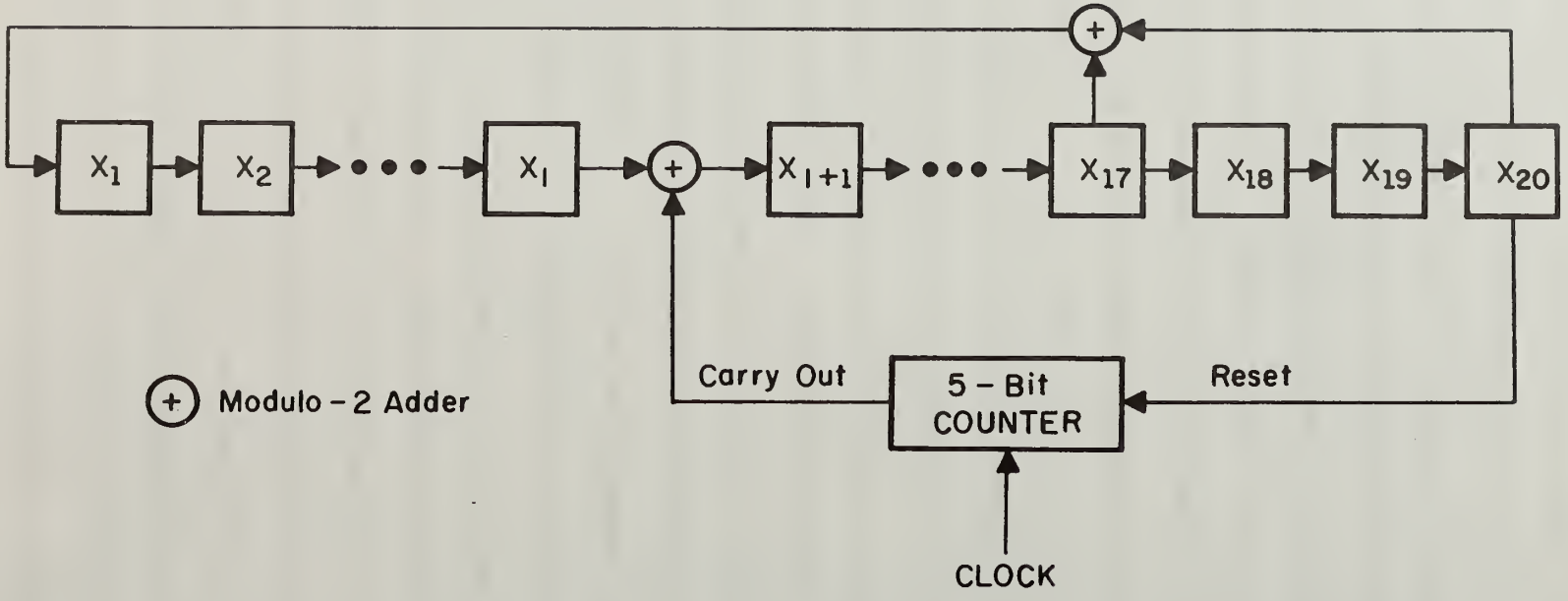


Figure 4.4 The Protective Circuit to Prevent the Feedback Shift Register from Staying in the All-Zero State

4.5 The Binary-Number-to-SRPS-Converter of the APE

The complete binary-number-to-SRPS-converter of the APE is illustrated in Figure 4.5. The entire converter employs a 20-stage shift register, a 5-stage counter, two 10-bit comparators, and two modulo-2 adders, better known as EXCLUSIVE-OR gates. The two 10-bit registers for storing the binary numbers are actually part of the input decoding circuits. Complementary Symmetry Metal Oxide Integrated circuits are used to implement this converter because of their extremely low power comsumption. The overall power consumption of the converter is less than 2 mW when it is operated at a 150 kHz clock rate. This power consumption level is much lower than any other approach could possibly attain. The dynamic performance of this converter is also very good. Whenever there is a change in the number to be converted into SRPS, the mean value of the output SRPS following the change and reach steady state in one clock period.

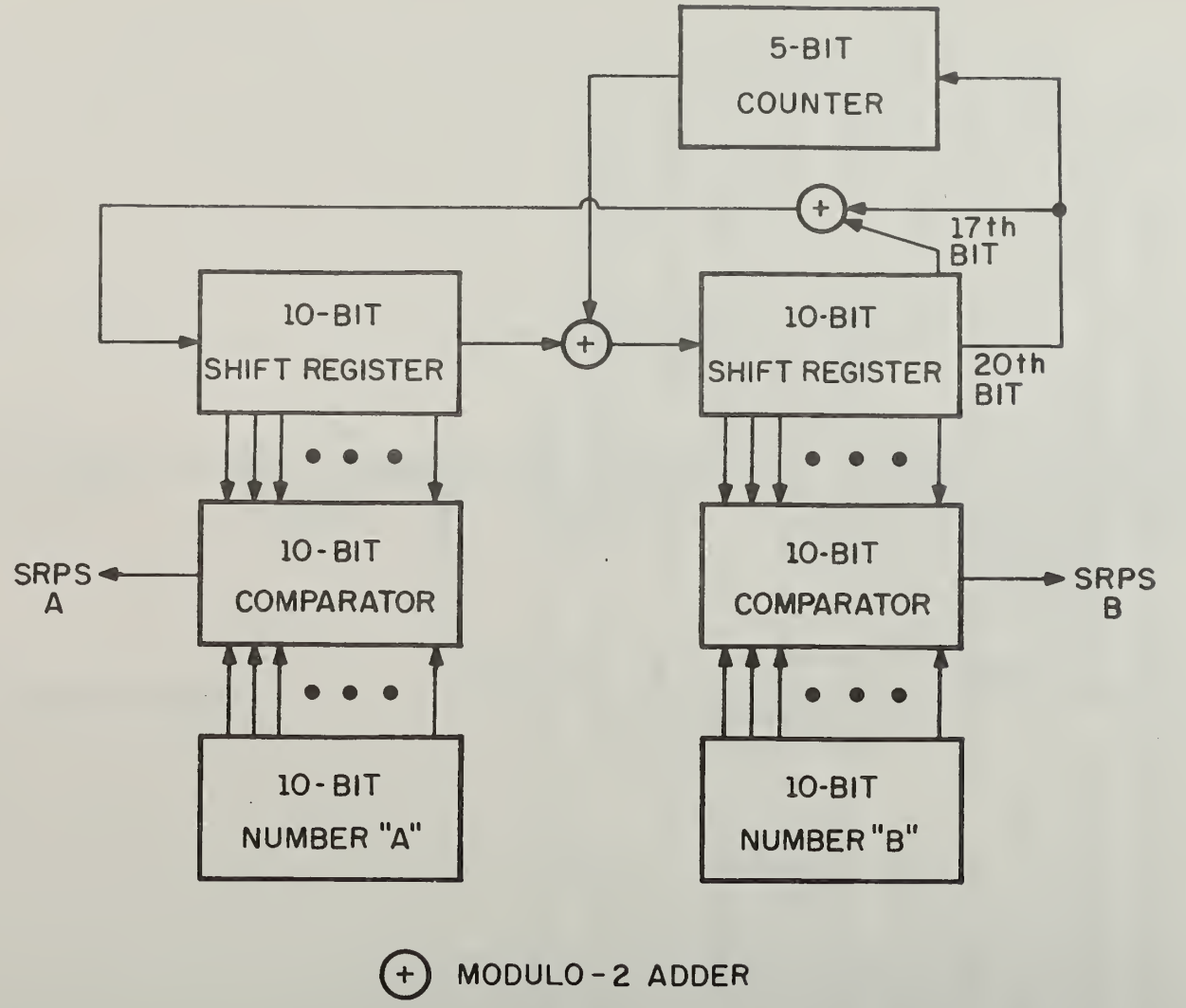


Figure 4.5 Conversion of Two Numbers into Statistically Independent SRPSs

5. CIRCUIT DESCRIPTION OF THE APEs

5.1 Special Design Considerations for the APEs

Because of the limited power available for the APE, the overriding factor in its design is low power consumption. An original estimate of the capability of remote powering of diverse systems puts the ceiling on the power consumption of an APE at 100 mW. This severe limitation on power consumption brings about many challenges in the implementation of the APEs and affects every facet of the circuit design. It immediately rules out the possibility of taking advantage of many commerically available communication circuits. In fact, all communication circuits used in the APEs were developed in the Digital Computer Laboratory of the University of Illinois. For logic circuits, COS/MOS integrated circuits are used exclusively because of their extremely low power consumption. In the design of the communication subsystem of the APE, multiplexing techniques are employed whenever possible to reduce the number of receivers and transmitrers required.

5.2 Block Diagram of the APEs

A functional diagram on an APE has been shown in Figure 2.3. It is presented in a way suitable to explain how the APEs function. However, in the implementation of the APE, some of the functional blocks are merged together while others are realized with separate physical hardware. From the hardware implementation point of view, the block diagram of an APE can be depicted somewhat differently as shown in Figure 5.1. There are two tunable receivers for data inputs. However one of these tunable receivers, namely receiver B in the figure, is operated on a time-sharing basis for both input data reception as well as program instruction reception. Furthermore, this

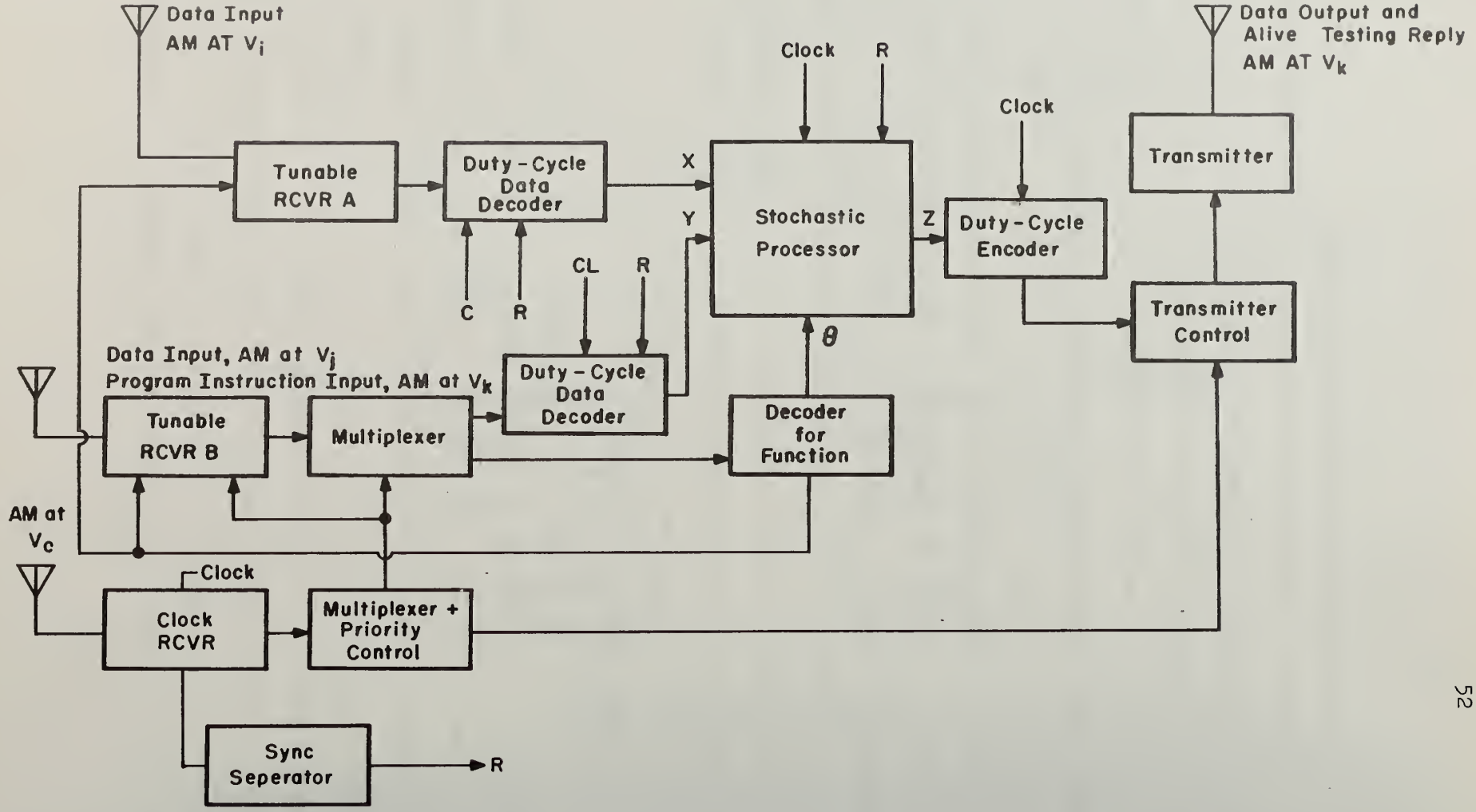


Figure 5.1 Block Diagram of an Autonomous Processing Element

receiver is operated at different priority levels with the reception of the program instruction having higher priority over the reception of input data. Whenever a program instruction is to be sent out from the program control unit, the latter is first switched to the programming mode. This causes the clock transmitter to stop the transmission of the clock signals. The absence of the clock signals, in turn, shuts off all output transmitters of the APEs. As a result the frequency channel for the program instruction transmission is clear from RF interference. The control of the tunable receiver B is then taken over for the transmission of program instructions and the receiver automatically tunes to the fixed frequency channel assigned to that particular type of APE for program instruction transmission, as denoted by ν_K in Figure 5.1. The same channel frequency ν_K is later used to transmit output data when the APE machine is in the execution mode. Because the APE transmitter is already shut off and no data output is sent during the programming mode, the channel is used exclusively for the transmission of the program instruction. The output signal of the time-shared receiver B is used to feed decoder for function during the programming mode and to feed the data decoder during the execution mode. The program instructions are sent, one at a time, to all types of APEs involved in a specific program. Upon the completion of the programming process, the program control unit is switched to the execution mode. Note that a program instruction contains the information of what channels the two tunable receivers are to be tuned to as well as what specific operation the stochastic processor is to perform! In the execution mode, the control of the tunable receiver B is returned to the APE for data input reception. Like the tunable receiver A, tunable receiver B is now tuned

according to the program instruction. A multiplexing technique is also employed to operate the transmitter, which is used to transmit the output data and the reply to the test signal from the program control as well. During the execution mode, the transmitter is used for output data transmission. It is otherwise used for replying to the test signal during the test mode.

The multiplexer control signal is sent to the APE from the clock channel. A continuous presence of an RF signal in the clock channel tells the APE to operate in the testing mode. For programming, as mentioned before, no RF signal is sent through the clock channel. In the execution mode, a sequence of RF pulses carrying the timing information is sent through the clock channel.

5.3 The Timing Circuit of the APEs

All operations of the APE are performed in synchronism with a common time reference. A composite clock carrying the clock signal and the synchronizing signal is sent from the APE control unit through the clock channel at 42.5 MHz. The clock signal has a frequency of 165.450 KHz and the synchronization signal uses 1 Hz. The APEs begin to communicate with each other at the occurrence of the synchronization signal. The communication period lasts $\frac{1}{9}$ of a second. Immediately afterwards, the computation begins and lasts for $\frac{8}{9}$ of a second. Then comes the next synchronizing signal and a new computing cycle begins. The waveform of a composite clock signal is illustrated in Figure 5.2. The clock signal is carried by the regular pulse while the synchronizing signal is carried by two consecutive wider pulses.

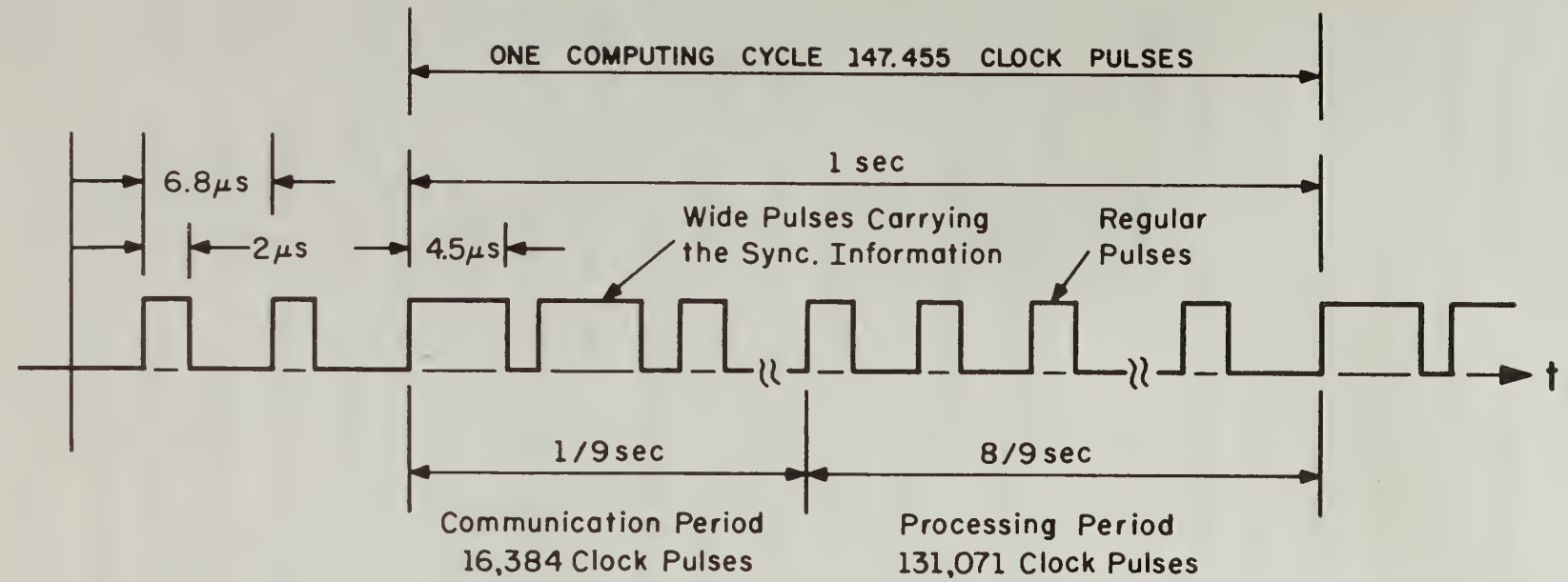


Figure 5.2 The Composite Clock Waveform

Three timing signals are required by the APE. They are the synchronizing signal, delayed reset signal and the preset signal. The synchronizing signal, also called reset signal, appears at the beginning of a computing cycle whereas the delayed reset appears one ninth of a second later and marks the end of data transmission and the beginning of computation. The preset signal occurs at about 3 msec before the next synchronizing signal does, and is used to set up the transmitter just before it is used to transmit the output data. The detail discussion on the generation of these timing signals is given below.

5.3.1 Digital Synchronizing Signal Separator

This part of the timing circuit separates the synchronizing signal from the composite clock. This is done by triggering a one-shot circuit with the composite clock. The output of the one-shot is a sequence of pulses with a fixed width which is wider than that of the clock pulse but narrower than that of the synchronizing pulse, as illustrated in Figure 5.3. The composite clock pulse sequence and the output pulse sequence of the one-shot are compared by feeding them into the data input and the clock input of a D-type flipflop respectively. The output of the D-type flipflop for the next clock period is equal to the value of the D input at the time the clock input C changes from a '0' to a '1'. By comparing the waveform of the composite clock and the output of the one-shot as shown in Figure 5.3, it is easy to see that the Q output of the flipflop is a '1' when the synchronizing signal occurs and is a '0' otherwise. Therefore the Q output of the flipflop contains only the synchronizing signal and is used for resetting the input storage at the beginning of every computing cycle.

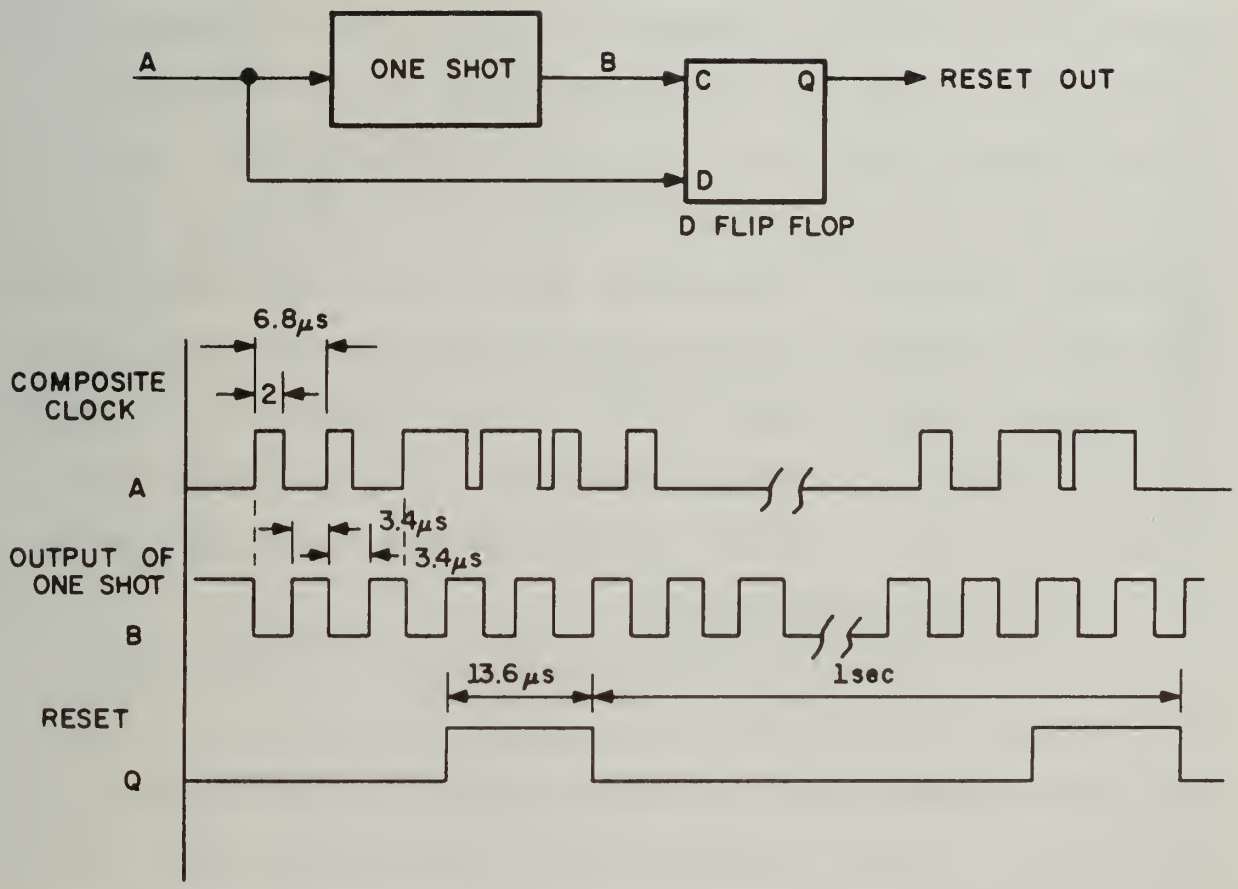


Figure 5.3 Synchronizing Signal Separator and its Waveform

5.3.2 The Delayed Reset Signal and the Preset Signal Generator

A delayed reset signal is required to signal the end of data transmission and the beginning of data processing. The delayed reset occurs 16,384 clock periods after the reset, or approximately $\frac{1}{9}$ of a second after the beginning of a computing cycle. The preset signal is used to turn on the output transmitter about 3 msec before the beginning of data transmission. The circuit used to generate these signals is shown in Figure 5.4. It employs an 18-bit counter with the reset input driven by the synchronizing signal. At the beginning of a computing cycle, the 18-bit counter is reset to the all-zero state, and the delay reset output is also set to '0' by the synchronizing signal. Then, the counter starts to count up. Once it reaches 16,384 counts, the output of the 15th bit changes from '0' to '1', causing the delay reset to change from '0' to '1' and to stay there afterwards, (due to the latch action of the S-R flipflop), until the next reset signal comes along. The preset signal is set to '0' at the beginning of a computing cycle. It changes to '1' when the counter counts up to 146,944, i.e. 511 clock periods or approximately 3 msec before the next reset pulse appears.

5.4 Mode Control Signal Detector

As mentioned earlier, the APE can be operated in three different modes, namely, the programming mode, the execution mode, and the test mode. During the programming mode, the output transmitter of every APE is shut off and a data input receiver is converted into the instruction receiver. For the execution mode, the output transmitter is active and the instruction receiver is converted back to an input data receiver. The signal to control the mode of operation is being transmitted to the APEs through the clock channel. For the execution mode, a composite clock is transmitted. In the

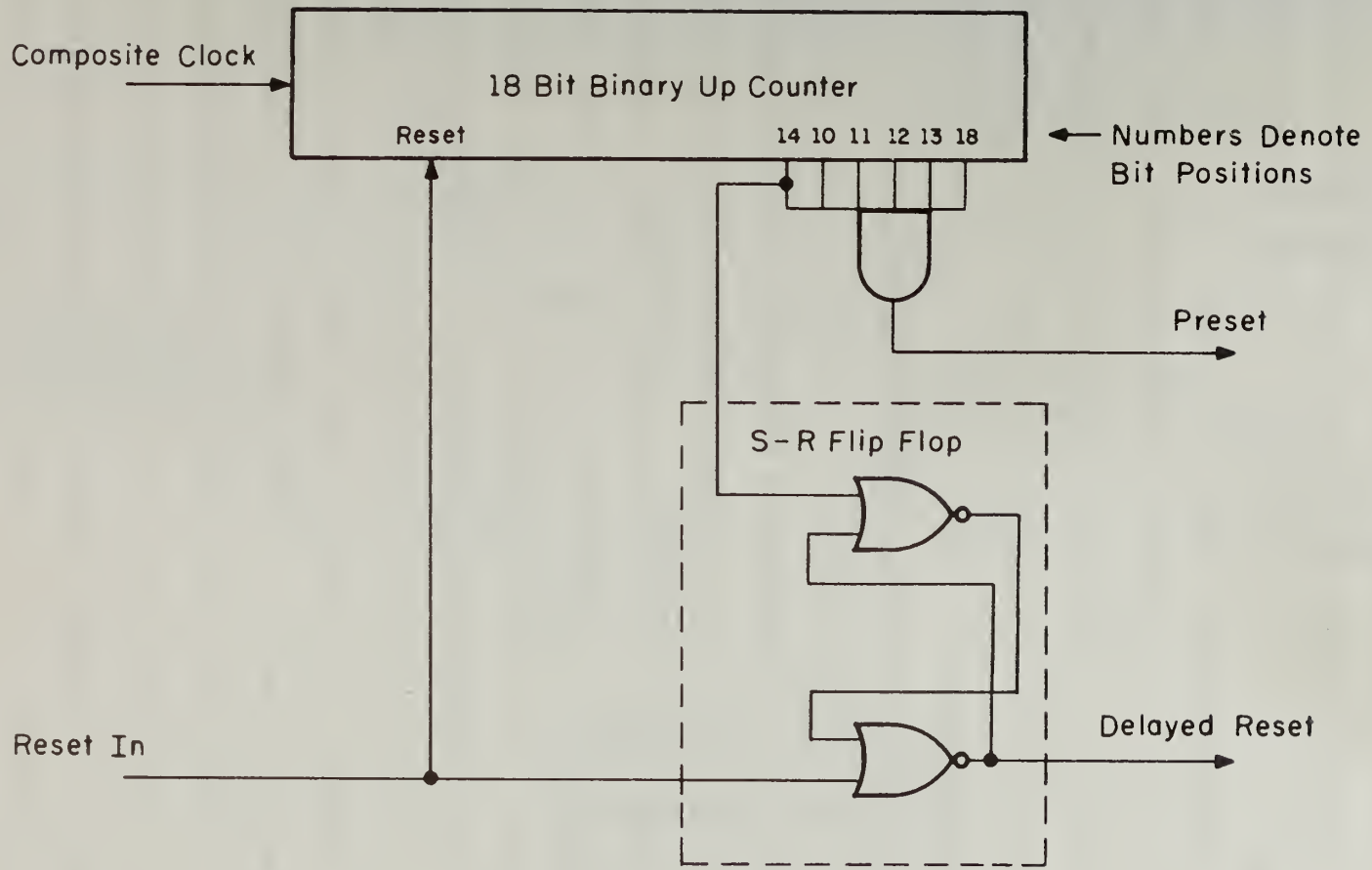


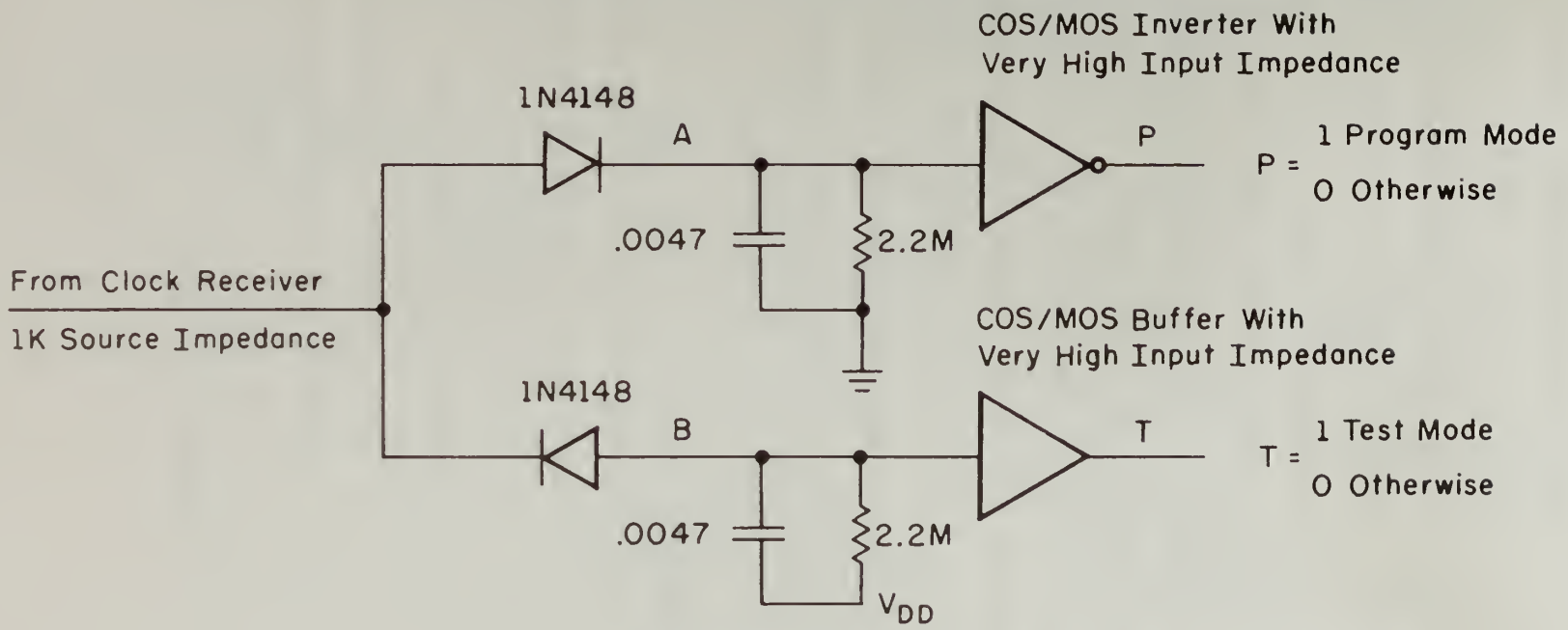
Figure 5.4 The Delayed Reset and Preset Signals Generator

case of the test mode, a continuous unmodulated RF signal is transmitted. For the programming mode, the clock transmitter is turned off completely.

The circuit for detecting these mode control signal is shown in Figure 5.5. The circuit employs two RC timing circuits controlled by two diodes separately. Point A can be charged to a logical '1' level much more easily than discharged back to a logical '0' level: As a result, point A is at a logical level '0' only during the programming mode. The output P is therefore at a logical '1' level only during this mode. By applying the similar argument, point T is visibly at a logical '1' level only for the test mode. The execution mode is identified by the logical '0' level in both T and P.

5.5 The Function-Decoder and Channel Multiplexer

The function of this decoder is to decode and store the instruction from the control unit. The instruction is coded in a binary pulse-width-modulated sequence 28 bits long, the first six bits carrying the operation code θ. The remaining 22 bits carry two tuning information sequences with 11 bits each used for the two tunable input data receivers. The decoding of this instruction is done with a network shown in Figure 5.6. When the APE machine is in the programming mode, the inverted P signal from the mode control signal detector is at a logical '0' level. This causes the signal at point A to be gated into point B, as well as setting the analog switch to the programming mode position. The resistor R in front of the COS/MOS NOR gate is used to perform noise discrimination. By adjusting the value of R, one could set the threshold on the input signal pulse width such that any narrower pulse will be rejected. (Normally noise induced by transient phenomena is in the form of narrow spikes!) After getting through the NOR gate, the instruction signal is decoded by comparing the width of each pulse



Mode Control Signals		
	P	T
Programing Mode	1	0
Execution Mode	0	0
Test Mode	0	1

Figure 5.5 The Mode Control Signal Detector

with that of a standard pulse in a sequence from the one-shot circuit. The comparison is accomplished by feeding the pulses into the data- and clock-inputs of a static shift register as shown in the diagram. When a narrow pulse in the instruction signal comes along, the data input of the register will be at a logical '0' level by the time the clock input changes from a logical '0' level to a logical '1' level. The register then receives a logical '0' input. When a wide pulse in the instruction signal comes along, a '1' is received. In this way, a sequence of pulse-width-modulated pulses is decoded and stored in the form of a sequence of binary numbers. The parts of the instruction carrying the tuning information are then sent to two non-linear D/A converters. This results in two analog tuning voltages custom-matched to the tuning characteristics of the individual tuning diode. An analog gate is employed to switch one of these analog tuning voltages to tune receiver B in the execution mode and to switch a fixed voltage for tuning the receiver B for instruction reception in the programming mode.

5.6 Input Duty-Cycle Decoder

Each APE has two input duty-cycle decoders, one each for input X and input Y. The function of these decoders is to convert the machine number in duty-cycle modulation into a 10-bit binary number and to store the result. The decoder consists of a 10-bit up-counter with a gated input. A clock signal is gated to the input of the 10-bit up-counter by the data input signal in duty-cycle modulation, as shown in Figure 5.7. Therefore, the data input is measured against the clock and the data input signal width is registered in the counter in terms of the number of periods of the clock.

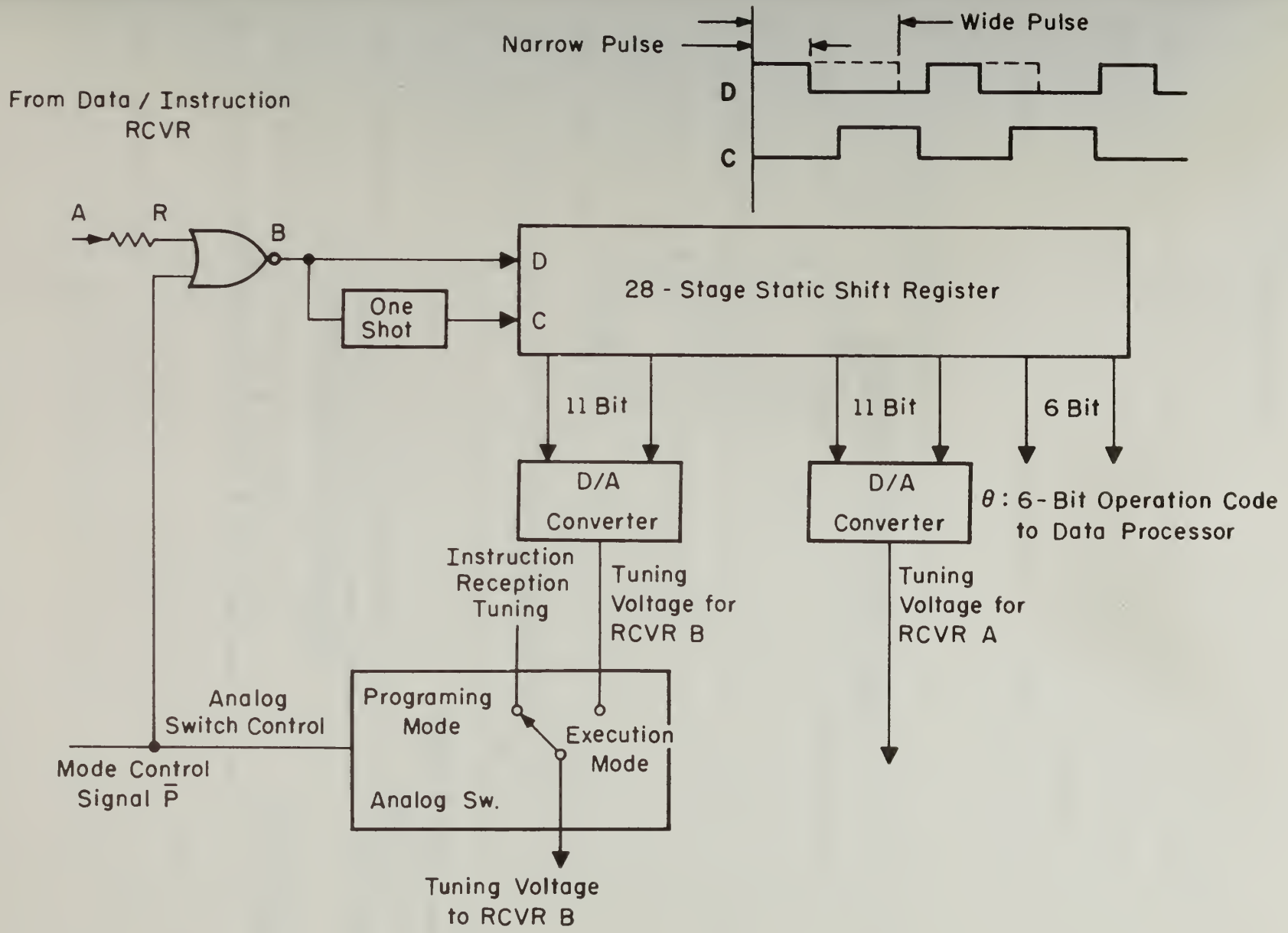


Figure 5.6 The Decoder for Functions and the Multiplexer

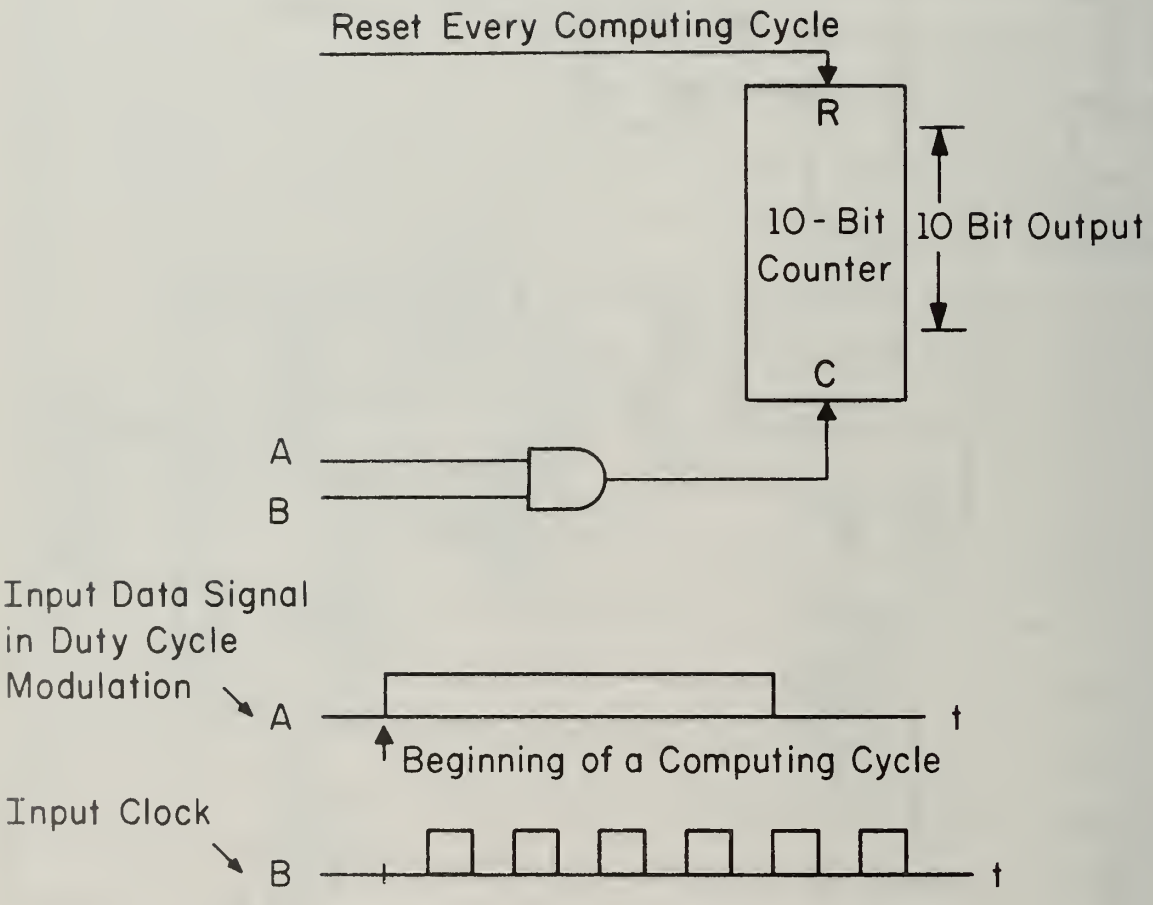


Figure 5.7 The Input Data Decoder

5.7 The Stochastic Processor

After the inputs X and Y are decoded and stored in binary numbers, they are processed by the stochastic processors according to the operation code θ from the instruction decoder. The stochastic processor consists of two binary-to-SRPS converters and the processing network, which have already been discussed in Chapter 3 and Chapter 4. They are not repeated here.

5.8 The SRPS Integrator and the Output Encoder

The function of the SRPS integrator is to convert the result of the stochastic processor in SRPS representation into binary representation for encoding into pulse width modulated signal. The integrator and the encoder are implemented with an up-down-counter as shown in Figure 5.8. The delayed reset changes from a logical '0' level to a logical '1' level at the end of $\frac{1}{9}$ of a computing cycle, triggering the one-shot circuit to send out a pulse. This pulse presets the counter according to the preset input: In the case of addition and subtraction, the preset input is a binary fraction of one half. It is otherwise all zero for multiplication and division. For storage the preset input equals the number to be stored. For any operation other than storage, the counter is set to count up (right after it is preset by the delayed reset pulse) and continues to do so until the end of the computing cycle. During the up-count period, it is driven by the output of the stochastic processor. At the end of the up-count period (which is also the end of the computing cycle), the content of the up-down-counter represents the result of the stocahstic processor in binary representation; it is ready to be encoded and transmitted at the beginning of the next

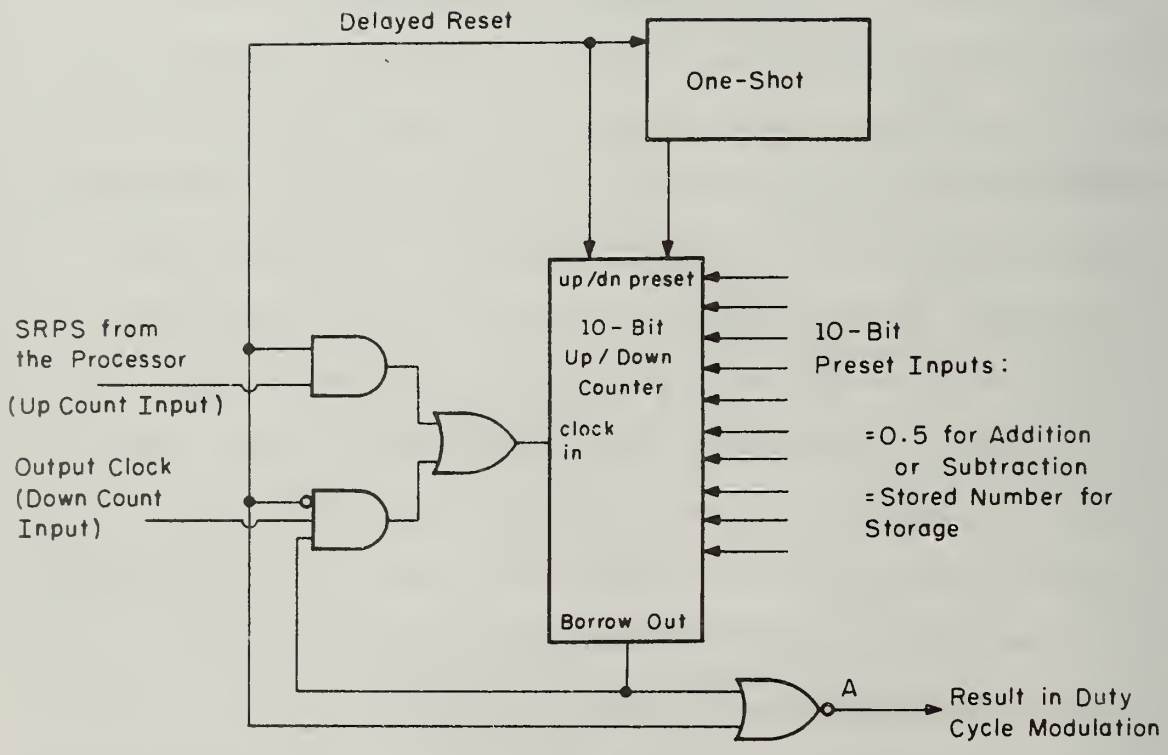


Figure 5.8 SRPS Integrator/Output Encoder

computing cycle. For the storage operation the up-down-counter is preset to the value of the number to be stored by the delayed reset pulse. This number remains in the counter until the end of the computing cycle.

At the beginning of the next computing cycle, the counter is set to count down. At the same time, the output clock, replacing the output of the stochastic processor, is used to drive the up-down-counter. The count-down operation continues until the counter is empty. It is obvious that the period over which the up-down-counter counts down corresponds to the initial number in the counter, which is the number to be stored in the case of storage operation and the estimate of the result of the stochastic processor in the case of other operations. Furthermore, the output represented by the length of the count-down period is already in the form of duty-cycle modulation. It can be used immediately to modulate the output transmitter.

5.9 The Communication Subsystem of the APE

The APE transmits and receives all types of information through radio frequency channels only. Each APE is equipped with two tunable receivers, a clock receiver and a transmitter for communication with other units of the APE machine. All these communication subsystems must be of the micropower type. This precludes the use of many commercially available communication circuits. They are built in fact entirely with discrete components. In the following sections, they are discussed in detail separately.

5.9.1 The Remotely Tunable Data Receiver

Although receiver design is thoroughly understood, it remains a very time-consuming process to design and develop a good one for a specific application. Of the numerous considerations given to the design of the

receiver, only the important ones shall be discussed below. The basic requirements of this receiver are as follows:

1) micropower consumption, with a maximum drain equal or less to 2.5 ma from a 6 volt source,

2) remote tunable feature for receiving pulse width modulated RF from any of the eleven channels covering a frequency band of 5.300 MHz,

3) a bandwidth of about 10 KHz to resolve a 100 msec input pulse,

4) a sensitivity of 500 microvolts,

5) a selectivity of at least "60 db down " for the nearest channel

Various types of receiving principles and techniques have been investigated for the implementation of this receiver. It was found that the heterodyne receiver is the most suitable because of its high sensitivity, high selectivity and simplicity in tuning. However, it is also well known that image frequency response and spurious response are the inherent shortcomings of this type of receiver. Although more sophisticated types of receiver design, such as superheterodyne and double conversion heterodyne types, could overcome these shortcomings, they would easily exceed the limit on the power consumption. Happily enough, proper reception with a heterodyne receiver is still possible if channel frequencies are assigned carefully, so that no channel frequency lies near a major spurious response frequency. Originally, it was planned to use a powerful microwave source at 1296 MHz to transmit the electric power to the APEs. To avoid interference, the frequency of the data channels are chosen to be as far down below the 1296MHz as possible. On the other hand, the range of tuning must be wide

enough to cover all channels. For a given tuning range, an increase in the frequencies of the band would decrease the tuning ratio and leads to simpler tuning circuits. Therefore it is preferrable to use higher channel frequencies from the tuning point of view. Other factors, such as the spurious frequencies and the availability of the components also affect the frequency assignment. A good compromise solution is to have the frequencies assigned between 15 MHz and 20.3 MHz, with the IF frequency at 10.811 MHz and with the channel spacing being 530 KHz. Within the tuning range of the readily available 10.7 MHz IF transformer for commercial FM receivers, the 10.811 MHz is chosen to be the IF because it minimized the interference of the spurious response. This can be best shown by examining the spurious response in more detail. Spurious responses are generated when harmonics of the input signal beat with the local-oscillator signal (or its harmonics) to produce a signal close to the intermediate frequency (or its subharmonics). To see how these higher harmonics mix, one could expand the transfer characteristics of the mixer transistor about the operating point by means of Taylor series

$$I_C = I_0 + \left(\frac{I_C}{\delta V_{BE}}\right) v_{be} + \frac{1}{2!}\left(\frac{\delta^2 I_C}{\delta^2 V_{BE}}\right) v_{be}^2 + \frac{1}{3!}\frac{\delta^3 I_C}{\delta V_{BE}^3} v_{be}^3 + \cdots \qquad 5.1$$

Where I_c denotes the total collector current

I_0 denotes the collector bias current

V_{BE} denotes the base to emitter voltage

v_{be} denotes the AC component of the base to emitter voltage.

Equation 5.1 can also be written as

$$I_C = I_0 + a_1 v_{be} + a_2 v_{be}^2 + a_3 v_{be}^3 + \cdots \qquad 5.2$$

where a_n is the n^{th} order coefficient in Equation 5.1.

In the ideal case of quadratic mixing, i.e. with $a_i = 0$ $i > 2$, involving two pure sinusoids, no spurious response could occur. However, for a practical mixer, the coefficients of higher degree are not zero, resulting in cross-modulation, modulation deepening and spurious responses. For a pulse width modulation scheme as in the case of the APE machine, the first two phenomena have less effect on the reception than the spurious responses. With the above-mentioned frequency assignment, the image response occurs in the band between 36.62 MHz and 41.92 MHz. These image frequencies are far away from any signal frequencies, they therefore cause no significant interference. The same is true for the first order subharmonic response. In this case, the response is caused by the higher even degree terms, such as the 4th, 6th, ... etc., as these terms cause the harmonics of the beat frequencies to appear. Let the wanted input signal be f_w; the spurious response, f_s; the IF frequency, f_i; and the local frequency, f_e, with

$$f_e = f_w + f_i \qquad 5.3$$

The first order subharmonic spurious response is given by

$$f_s = f_w + \frac{f_i}{2} \qquad 5.4$$

When this f_s beats with f_e, the result is $\frac{f_i}{2}$, the first order subharmonic of the f_i. According to Equation 5.4, this type of spurious response for $f_w = 15.00$ MHz is 20.405 MHz. The frequencies of this type of spurious response for higher frequency channels are of course higher in frequency. Therefore, they all lie outside the active band of frequencies and cuase no significant interference!

For higher order but less significant spurious response, it is not possible to eliminate all unwanted signals completely from the active band.

However, a properly chosen IF frequency places the next most significant spurious response as far away from data channels as possible. The next most significant spurious response is the second order image response. It is caused by the second harmonics of the unwanted signal beating with the local frequency to produce f_{IF}. For example, let f_w = 15.00 MHz, f_{IF} = 10.811 MHz, the local frequency is therefore f_e = 25.811 MHz. Now if an unwanted signal exists at frequency f_s = 18.311 MHz, its second harmonic beating the local frequency will produce exactly the intermediate frequency. If f_s = 18.311 MHz happens to be another data channel frequency transmitting data at much higher power level, harmful interference could occur. The intermediate frequency obviously affect the location of these spurious response. The optimum IF frequency can affect the location of these spurious responses. The optimum IF frequency can be shown to be 10.811 MHz. Figure 5.9 shows different type of spurious response.

The electronic tuning of the receiver is done by means of a reverse-biased diode. It is well known that the junction capacitance of a reverse biased diode is given by

$$C = \frac{C_0}{(1 + \frac{V}{\phi})^{\xi}} \qquad 5.5$$

where C_0 is the capacitance at zero bias

V is the reverse-biased voltage

ϕ is the contact potential

ξ is a constant determined by the impurity gradient of the diode (for an abrupt junction $\xi = \frac{1}{2}$)

The tuning voltage ranges from 1 volt to 6 volts is supplied from the decoder for function as described before. The tuning circuit and the other part of the receiver are shown in Figure 5.10.

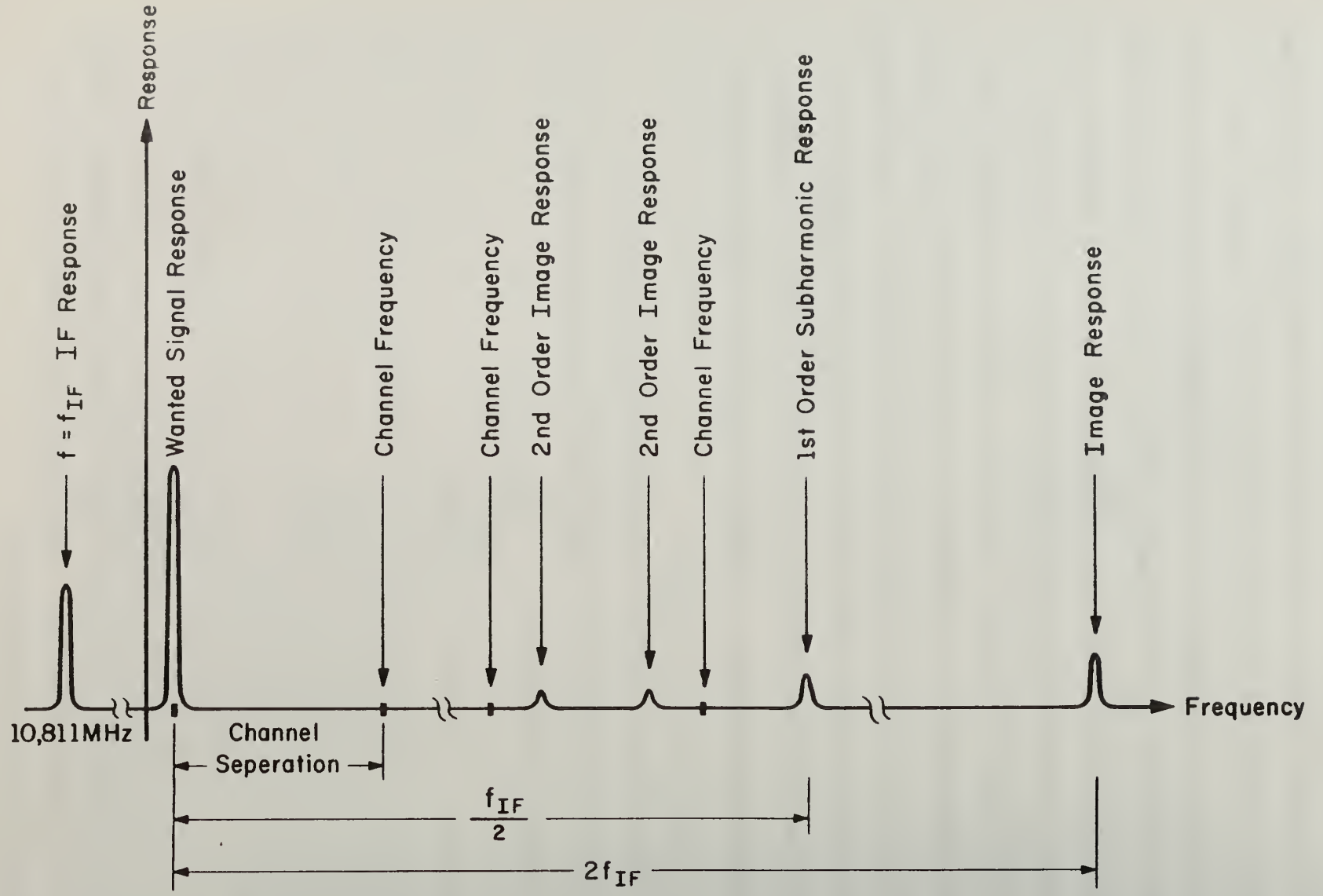
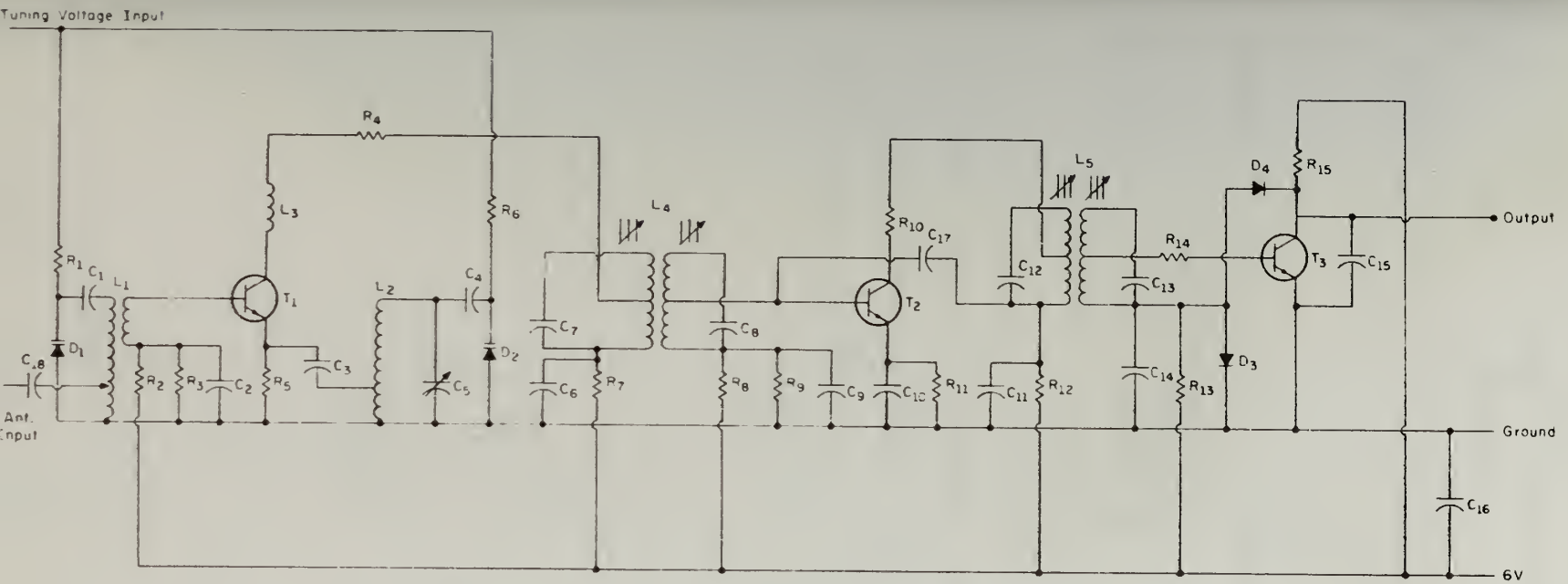


Figure 5.9 Some Common Spurious Responses of Heterodyne Receiver



C_1 = 0.0047μF
C_2 = 0.0047μF
C_3 = 560 MICA
C_4 = 0.0047μF
C_6 = 1μF
C_7 = 50PF
C_8 = 50PF
C_9 = 0.0047μF
C_{10} = 1μF
C_{11} = 1μF
C_{12} = 50PF
C_{13} = 50PF
C_{14} = 0.0047μF
C_{15} = 270PF
C_{16} = 2.7μF
C_{17} = 4.7PF NP750
C_{18} = 27PF
C_5 = 3PF - 12PF
R_1 = 27KΩ
R_2 = 47KΩ
R_3 = 10KΩ
R_4 = 220Ω
R_5 = 680Ω
R_6 = 27KΩ
R_7 = 220Ω
R_8 = 47KΩ
R_9 = 10KΩ
R_{10} = 220Ω
R_{11} = 220Ω
R_{12} = 220Ω
R_{13} = 27kΩ
R_{14} = 470Ω
R_{15} = 15KΩ
L_1 = Heat GD-19-40-914
L_2 = 5.4 Turns on DALE IPB-2041-31 Form, Tapped at 2.2 Turns
L_3 = 5.2 Turns on the Same Coil Form of L_2 but Opposite Direction
L_4, L_5 = Miller 8851-A
T_1, T_2, T_3: RCA 40245
D_1 = MV2115
D_2 = MV2109
D_3, D_4 = 1N4148

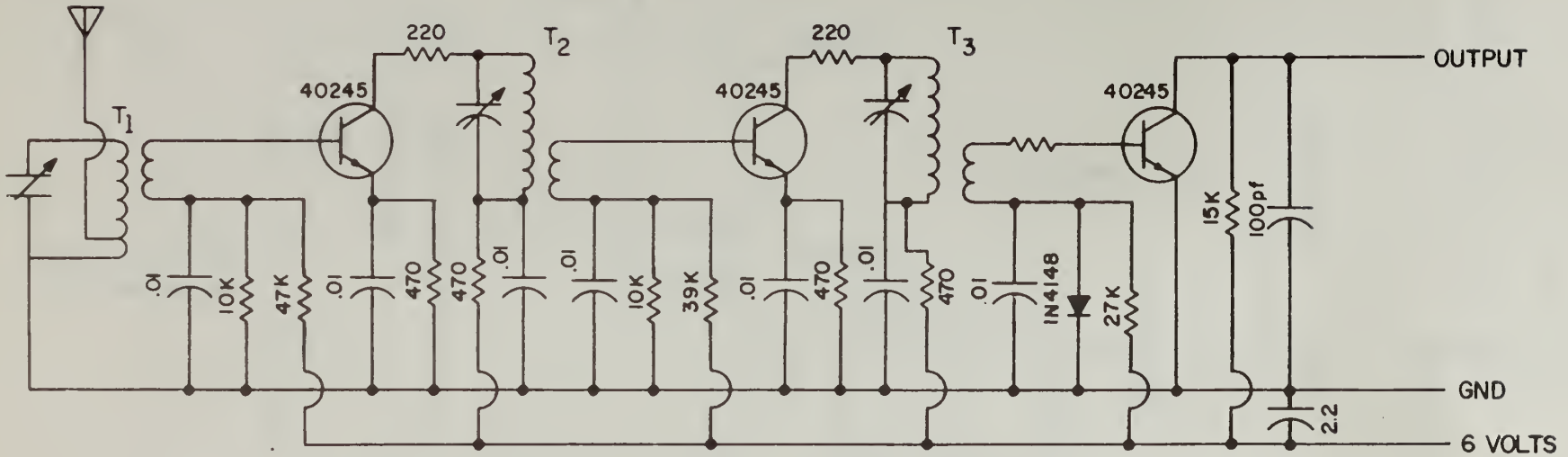
Figure 5.10 Remotely Tunable Heterodyne Receiver

5.9.2 The 42.5 MHz Clock Receiver

This receiver consist of a 2-stage tuned RF amplifier followed by an AM detecting stage. The total power consumption is 3 ma from a 6-volt source. The sensitivity is 200 microvolts,and its bandwidth is about 1 MHz. The circuit diagram of this receiver is given in Figure 5.11.

5.9.3 The Switching Transmitter

The transmitter is shown in Figure 5.12. It is a crystal-controlled low power switching transmitter developed in the Computer Laboratory to satisfy the particular requirements of the APE machine. It is a tune-gate and tune-drain oscillator. The second gate of the dual-gate FET is used to switch the gain to turn on or turn off the oscillator. The transistor 2N2475 is used to reduce the standby power consumption while the 2N2894A is used for cutting down the Q value of the tank circuit when the transmitter is turned off, i.e. in order to produce a RF pulse with a sharp decay. The total power consumption at 15% duty cycle is about 6 mW.



T_1 : HEATH GD-19-40-916 TRANSFORMER

T_2 and T_3 : DALE IPB-2042-31 RF TRANSFORMERS

Figure 5.11 The 42.5MHz Clock Receiver

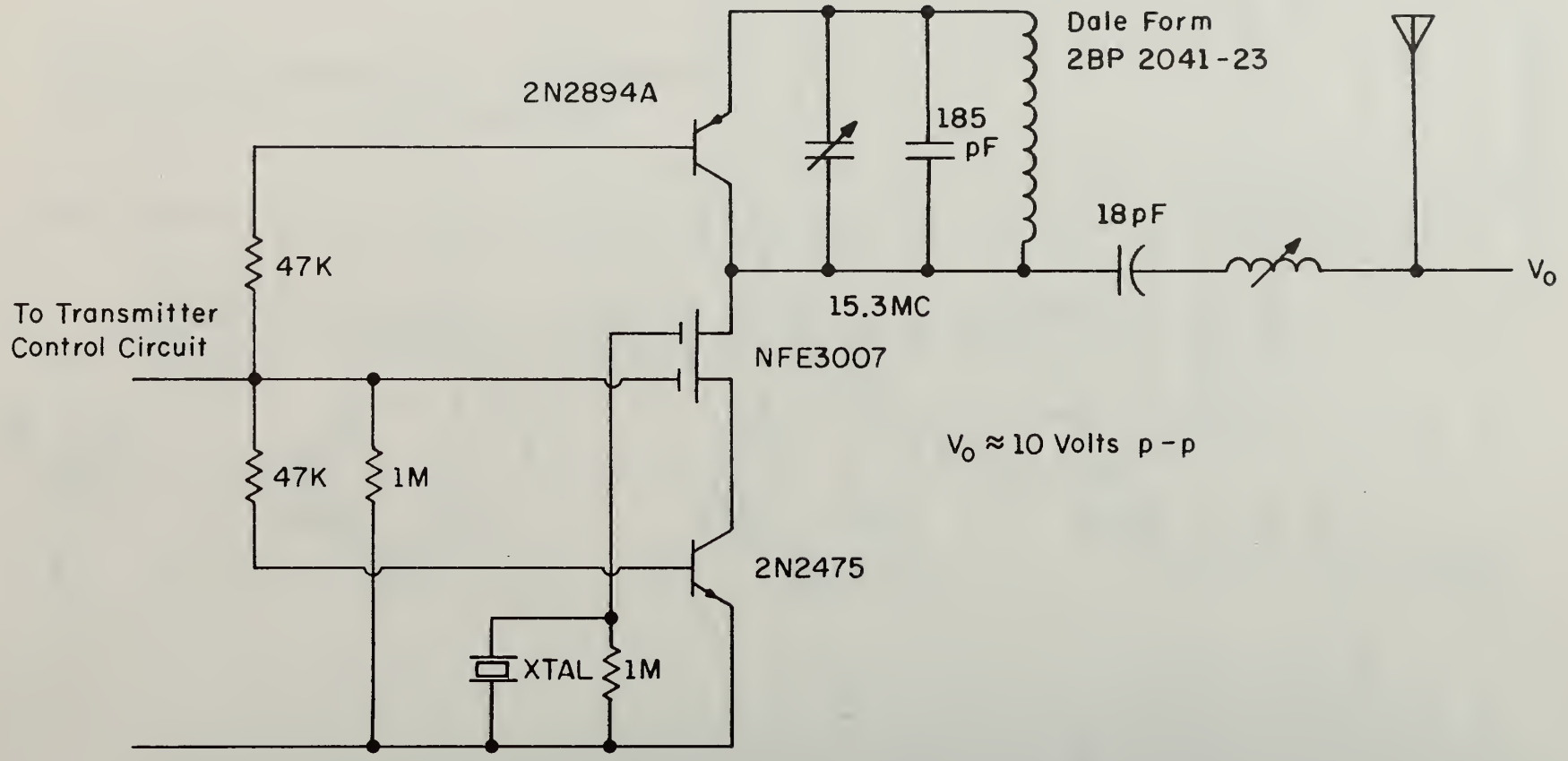


Figure 5.12 The Switching Transmitter of the APE

6. THE APE CONTROL UNIT

6.1 Block Diagram of the APE Control Unit

The block diagram of the APE control unit is shown in Figure 6.1. It can be divided into three major functional units. The first one is the program instruction encoder and transmitter. The program instructions are put together by the settings of the operation selector and of the channel selectors for any type of operation, except storage. In the case of storage, the tuning information is not needed. Instead, the number to be stored has to be incorporated into the instruction. This number is chosen in signed decimal representation by the thumbwheel switch and converted into a 10-bit machine number in binary representation. The multiplexer, controlled by the operation selector, then puts this 10-bit binary number into the instruction code generator, instead of the information from the channel selector as in the case of other operations. The instruction code generator is used to encode the instruction into a sequence of pulse-width-modulated pulses. This sequence of pulses is then used to modulate the carrier frequency of the channel over which the instruction is to be transmitted. The second functional unit is the clock generation and transmission unit. It consists of a composite clock generator driven by a master oscillator and a 42.5 MHz transmitter. The third functional unit is the output monitoring, decoding and display unit. It consists of an all-channel receiver, a machine-number-to-BCD converter, a scaler, a Nixie display unit, an "alive indicator", and finally an ASCII encoder for interfacing with a teletype. The output of any APE channel can be monitored by the all-channel receiver. If the APE machine is in the test mode, the signal receiver is used to drive the alive indicator. In the execution mode, the

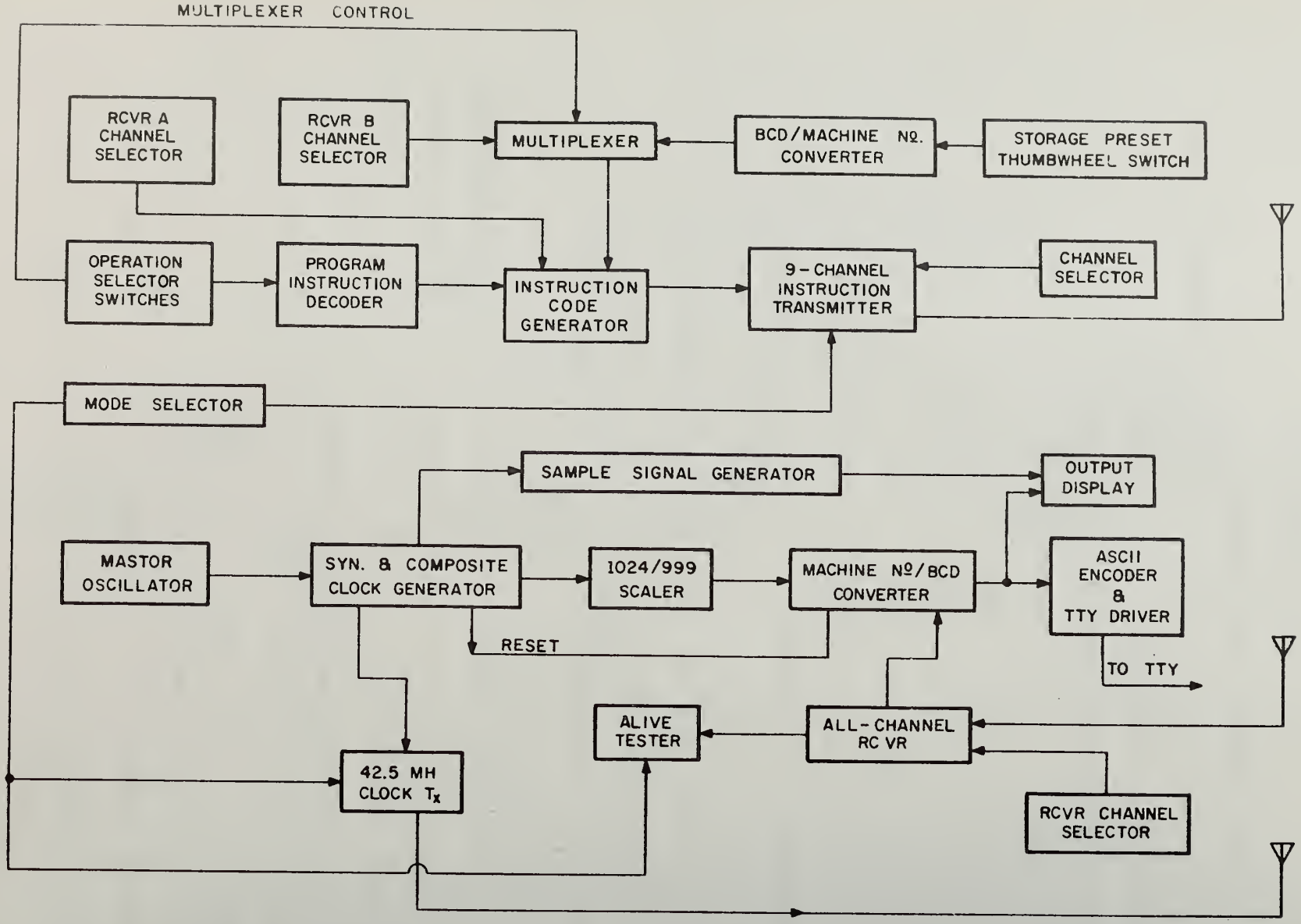


Figure 6.1 A Block Diagram of the APE Control Unit

received signal is the output data from the APE of that channel. This data is in machine numbers and is converted into signed BCD codes for decimal display. The 1997/1023 scaler is to scale a 10-bit binary machine number into three decimal digits plus sign. In the following sections, detailed discussion will be given to the non-trivial functional blocks.

6.2 The Instruction Code Generator

It consists of a pulse sequence generator, a counter, a multiplexer, and a monostable multivibrator as shown in Figure 6.2. The pulse sequence generator produces a sequence of 32 pulses when a starting signal is received. The counter and the multiplexer are connected to form a parallel-to-serial converter. The instruction word is fed to the data inputs of the multiplexer. As the counter starts to count, the inputs to the multiplexer are gated sequentially to the output. The output of the multiplexer is used to modulate the timing network of a monostable multivibrator by switching an auxiliary timing resistor in and out of the timing network. The output from the monostable multivibrator produces therefore a binary-pulse-width-modulated signal. Figure 6.3 shows the actual circuit of the instruction code generator. It should be pointed out that the instruction code generator is designed to handled instructions up to 32 bits long. (For the APE machine, the instruction is only 28 bits long!)

Figure 6.3 shows the actual circuit of the instruction code generator. Two SN74150N's are employed to form the 32-bit multiplexer. The diode D is used to switch the 4.7K auxiliary timing resistor R_T in and out of the network. The ready indicator circuit is to turn on an indicator light when the instruction code generator has completed sending out a full instruction

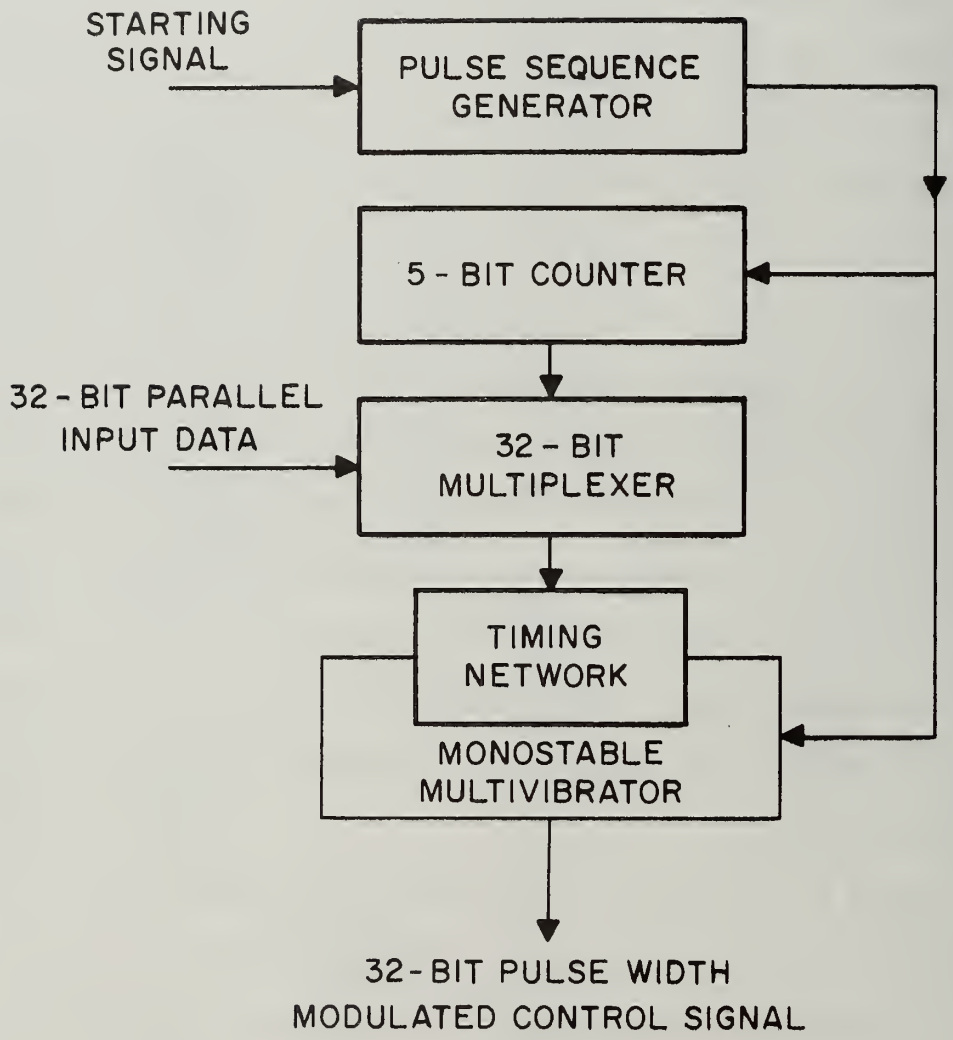
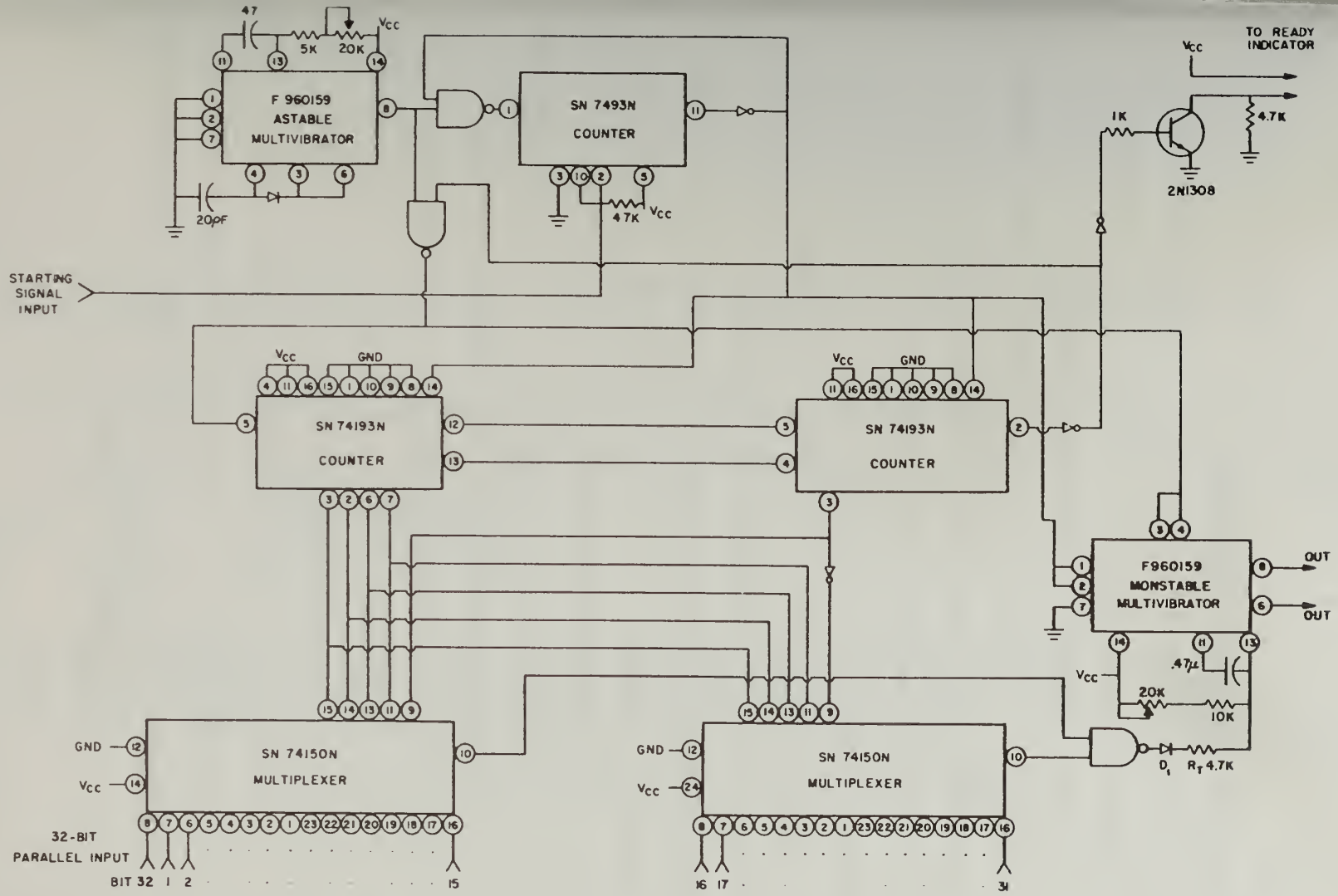


Figure 6.2 A Block Diagram of the Instruction Code Generator



NUMBERS IN CIRCLE INDICATE THE PIN NUMBER OF I C's

Figure 6.3 Schematic Diagram of the 32-Bit Instruction Code Generator

word and is ready for loading for another word in another APE channel. Figure 6.4 shows an instruction word and the corresponding pulse-width-modulated signal.

6.3 The 9-Channel Instruction Transmitter

This transmitter consists of nine separate crystal-controlled RF oscillators and modulators. Whenever an instruction is to be transmitted over a specific channel, the power to the corresponding RF oscillator and modulator is switched on. Otherwise, their power supplies are switched off. This simplifies a great deal of the RF shielding problem because the RF signals of the instruction transmitter are generated only when they are needed during the programming mode. The schematics of the transmitter for each channel is given in Figure 6.5: It is essentially a gated RF source. The crystal-controlled oscillator generates the carrier which is sent through a tuned amplifier to remove the harmonics and is amplified to about one volt in amplitude. Then it is fed to the first gate of a dual gate FET. The second gate of the FET is switched between +8.5 volts and -5 volts. When it is held at +8.5 volts, the dual FET is operated in the active region. The RF signal will be further amplified before it is transmitted. When the second gate is at -5 volts, the FET is switched off, allowing only very small amounts of RF to leak through. The reason for using the dual gate FET is its extremely small gate-to-drain capacitance which leads to a better on-off ratio in switching the RF signals than would be possible with its bipolar equivalent. The switching voltage is provided by the switching circuit of the transistors 2N1308 and 2N1309. The diode in the emitter of the 2N1308 is used for improving the noise margin of the input.

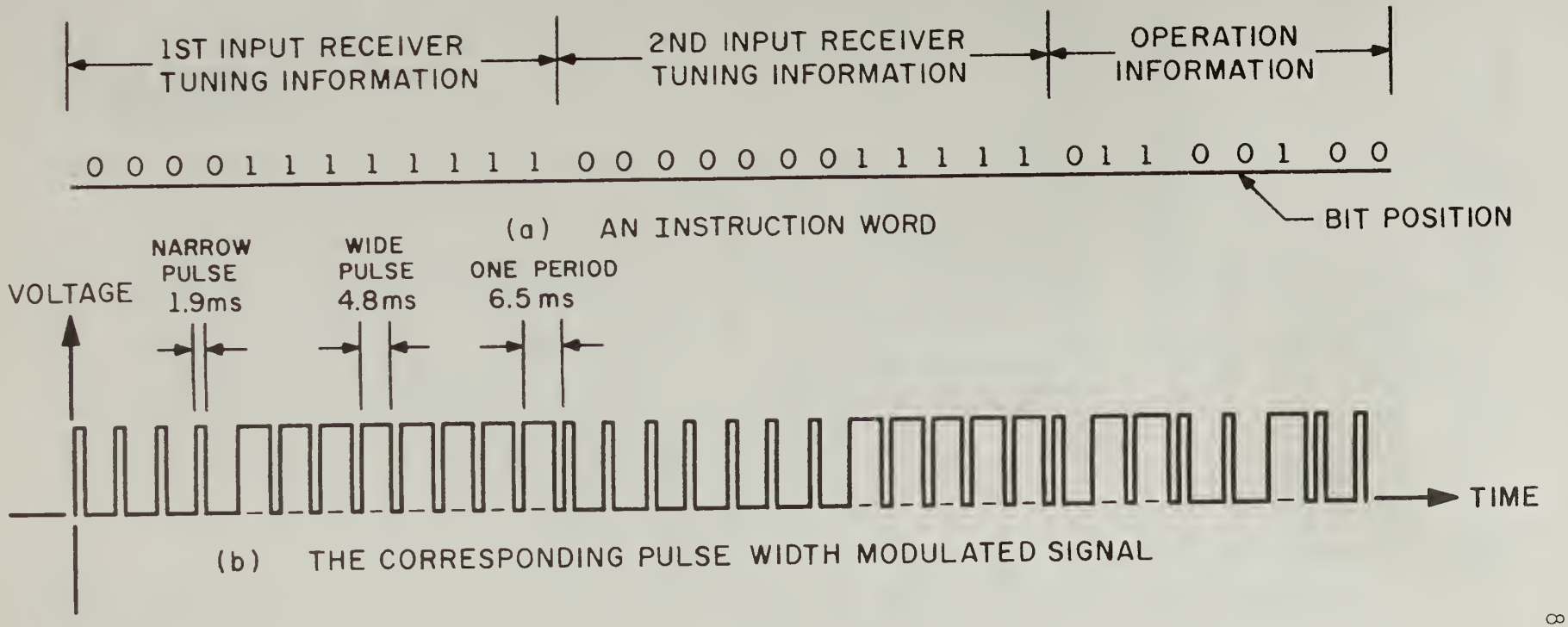


Figure 6.4 An Instruction Word and the Corresponding Pulse Width Modulated Signal

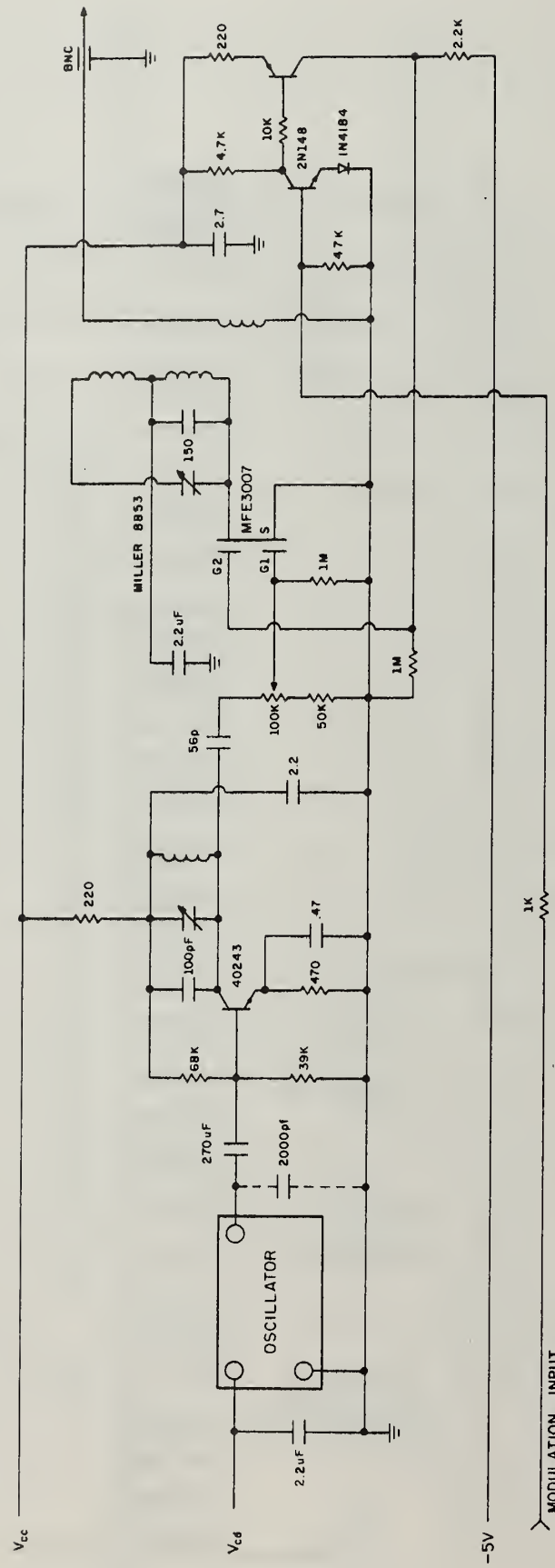


Figure 6.5 The Instruction Transmitter for an APE Channel

6.4 The Clock Signal Transmitter

The 42.5 MHz clock transmitter has the same schematics as the instruction transmitter but is operated at higher frequency. However, the clock transmitter operates during the execution mode. In order to avoid interference with other circuits, it is physically built in a shielded box. The operation principles of the clock signal transmitter are identical to those of the instruction transmitter discussed in Section 6.3. It is not repeated here.

6.5 The BCD/Machine Number Converter

The inputs to this converter are three decades of BCD code together with a sign bit, all from the thumbwheel switch in the front panel. The output is a 10-bit machine number. The conversion is accomplished in two steps. The first step is to convert the signed BCD number into a signed binary number. The second step is to transform, according to Equation 3.8, the signed binary number into a machine number. The first step is implemented with six 4-bit BCD/binary converters SN74184, connected as shown in Figure 6.6. The transformation in the second step corresponds to inverting the binary number after shifting it one bit position to the right. The inversion is done by nine EXCLUSIVE-OR gates acting as TRUE/COMPLEMENT gates. Shifting is done simply by wiring the output binary bits into the shift position. Note that in the machine number, a negative number can be obtained by inverting a positive number with the same magnitude. Therefore, if the sign bit is used to control the TRUE/COMPLEMENT gates, as shown in the figure, the result will be the machine number equivalent to the 3-digit signed decimal number of the inputs.

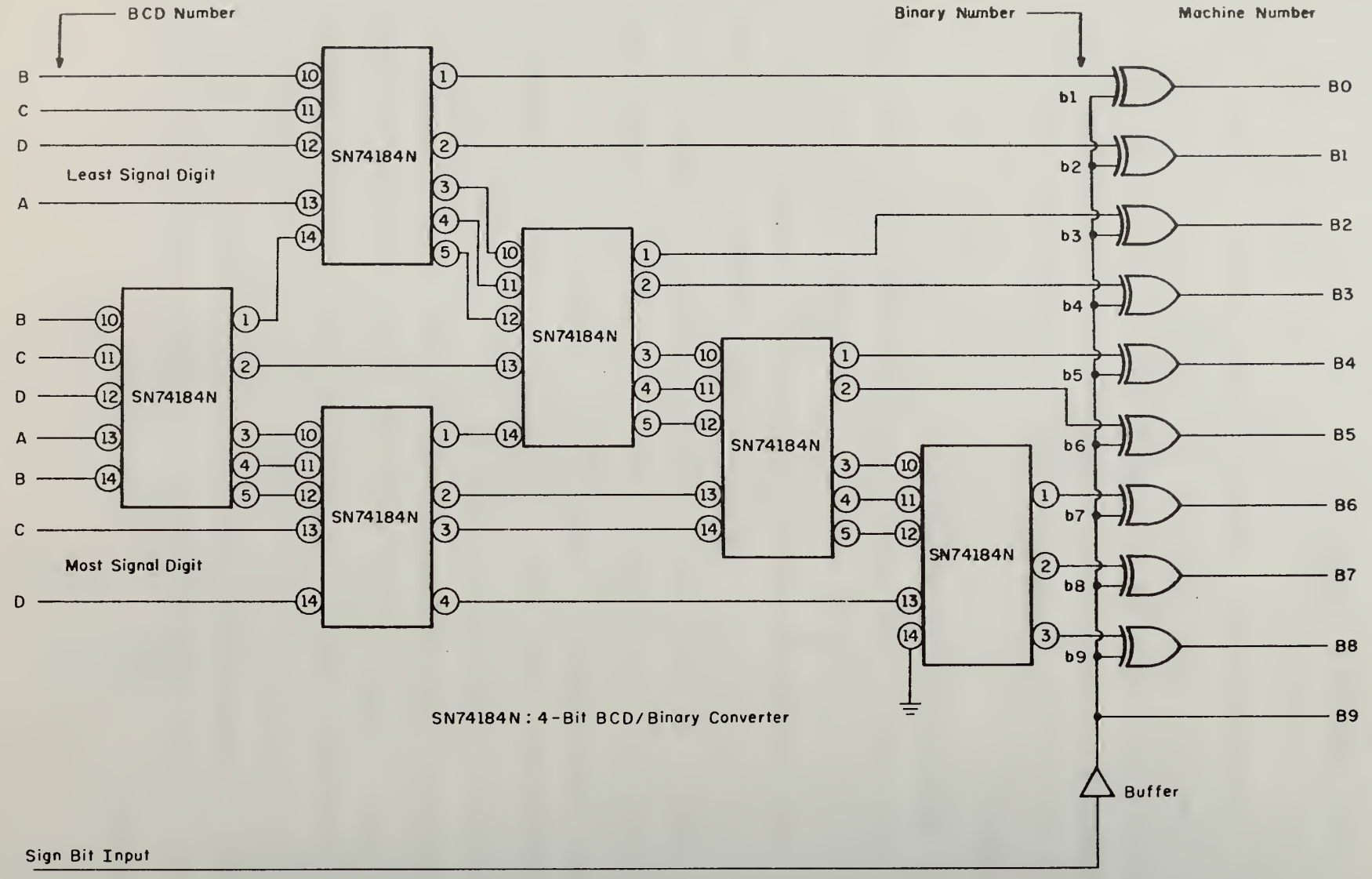


Figure 6.6 The BCD/Machine Number Converter

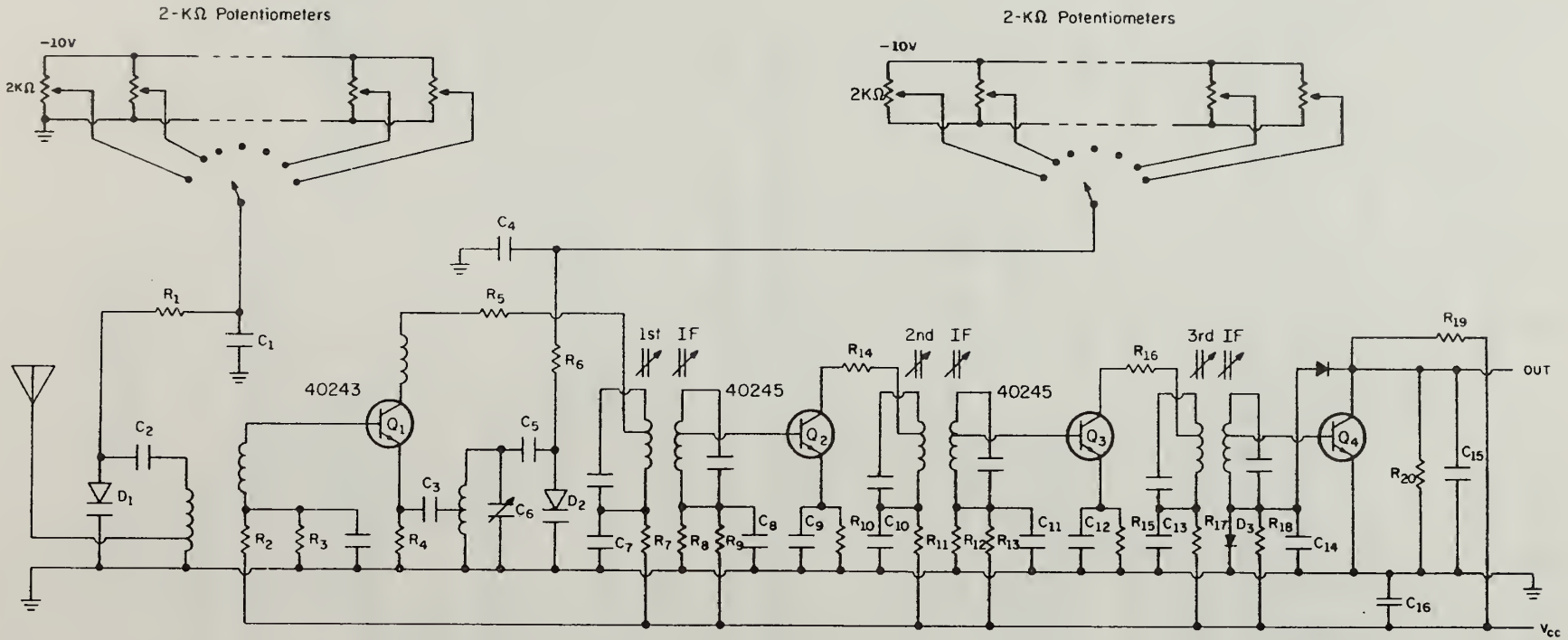
6.6 The All-Channel Receiver

The receiver is used to receive the output data signal as well as the test reply signal from the APEs. It is a heterodyne type AM receiver, as shown in Figure 6.7. Its operation is similar to the one used for receiving input data in the APE, except that this all-channel receiver has an extra IF stage: Its sensitivity is then about 30 microvolts. It is housed in a shielded box to reduce the interference from the nosiy environment of the TTL logic networks. The tuning is done with twelve pairs of potentiometers. Each pair of potentiometers is adjusted to produce a pair of voltages for tuning the RF input resonant circuit and the local oscillator resonant circuit for the reception of a particular channel. A particular pair of voltages is switched into the tuning circuits whenever the particular channel is to be tuned in.

6.7 The Machine Number to BCD Converter and the 1997/1023 Scaler

After the input data to the APEs are processed, the results are sent out by the APEs in machine number representation. For easy read-out of the results, it is necessary to have them converted back to signed decimal representation. This is done in the control unit by the 1997/1023 scaler together with the machine-number-to-BCD converter.

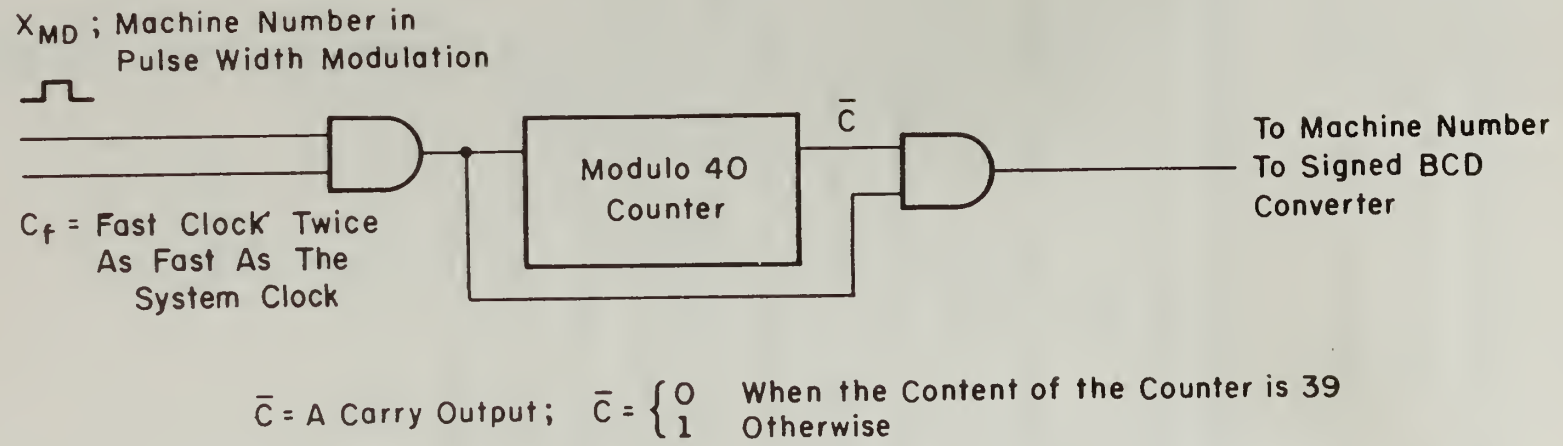
The output signals from the APE are in pulse-width modulation. The pulse width ranges from 0 to a maximum of 1023 clock periods. The 1997/1023 scaler is to linearly map a number in this range into a number ranging from 0 to 1997 (to the limit of the truncated least significant digit ...) Figure 6.8 shows the block diagram of the scaler. What it does is simply to delete one pulse for every forty input pulses. A fast clock having a frequency twice as



R_1-27K	R_6-27K	R_{11}-220	R_{16}-220	C_1-.068μf	C_6-adj.	C_{11}-1μf	D_1-MV2115
R_2-220	R_7-27K	R_{12}-10K	R_{17}-220	C_2-.0047μf	C_7-1μf	C_{12}-1μf	D_2-MV2109
R_3-10K	R_8-10K	R_{13}-47K	R_{18}-27K	C_3-75pf	C_8-1μf	C_{13}-1μf	D_3-1N4148
R_4-690	R_9-47K	R_{14}-220	R_{19}-15K	C_4-0.68μf	C_9-1μf	C_{14}-1μf	
R_5-220	R_{10}-220	R_{15}-220	R_{20}-100K	C_5-4700μf	C_{10}-1μf	C_{15}-1000pf	
						C_{16}-.033μf	

Figure 6.7 The All-Channel Receiver of the APE Control Unit

$0 < X_{MD} < 1023T_S$; T_S = Period of System Clock



$\bar{C}$ = A Carry Output; $\bar{C} = \begin{cases} 0 & \text{When the Content of the Counter is 39} \\ 1 & \text{Otherwise} \end{cases}$

Figure 6.8 The Block Diagram of the 1997/1023 Scaler

high as the system clock is used in the input of the scaler for comparison with the pulse-width modulated signal from the output of the APEs. With such an arrangement, the number of the output pulses is that of the input, scaled by a factor 1997/1023.

After the scaling, an inverse transform of Equation 3.8 is performed by the machine-number-to-BCD converter as shown in Figure 6.9. The decade up-down counters are preset to 999 at the beginning of a new computing cycle, just before the counter starts to count down. When the counter reaches the all zero state, it is switched to count up. How long the counting process lasts depends on the output signal from the APE channel to be displayed. If the counting process ends during the count-down period, the sign of the result is positive. Otherwise it is negative. In both cases, the magnitude of the result is given by the value of the final count. The full range of the input to this converter is 1997 pulses. This corresponds to an output range from -999 to +998. Therefore, the combined function of the scaler and the converter will transform a 10-bit number into a 3-digit signed decimal number to within the accuracy of the 10-bit number.

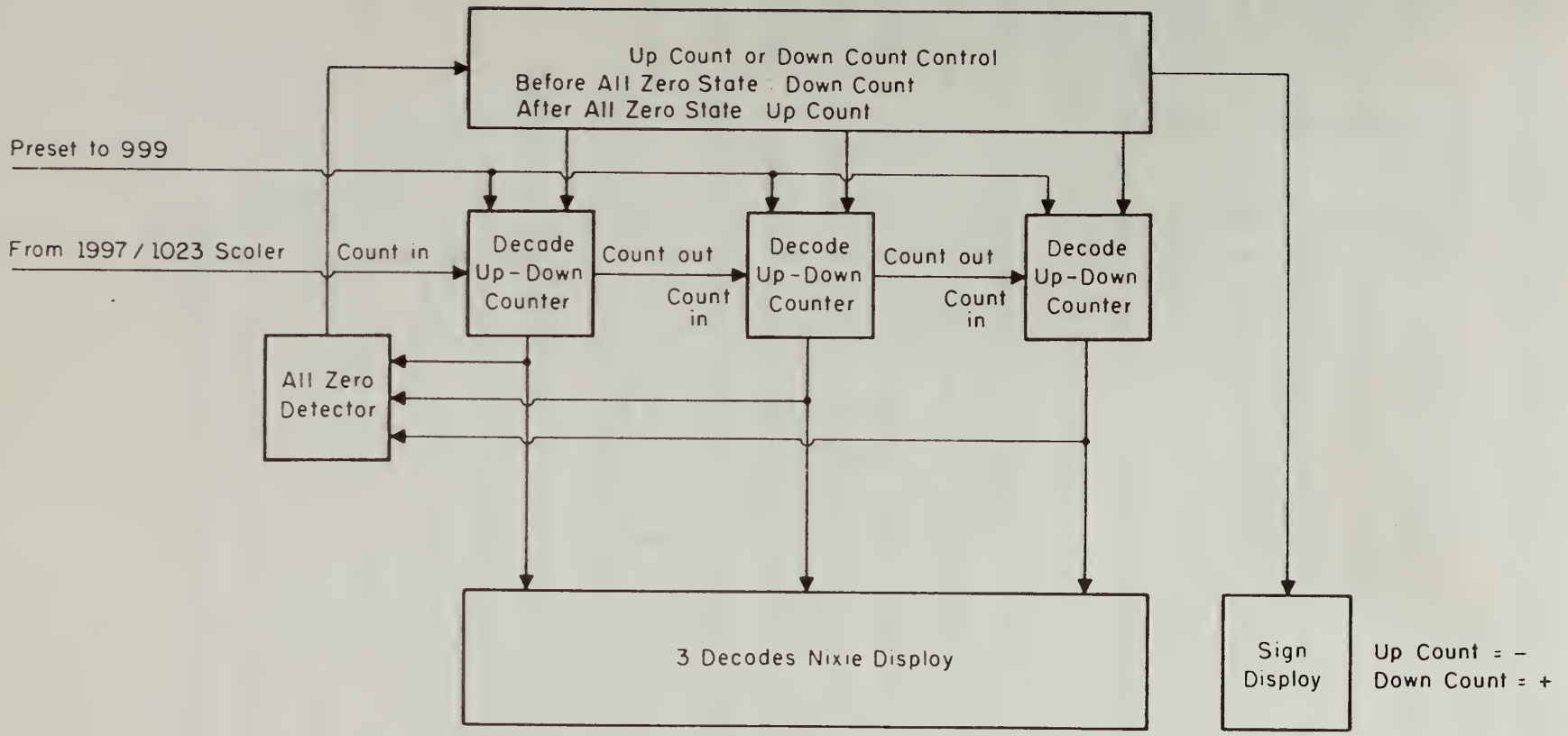


Figure 6.9 The Block Diagram of the Machine-Number-to-BCD Converter

7. THE APE SENSORS AND THE REMOTE POWER SUPPLY OF THE APEs

7.1 The APE Sensor

For input data acquisition, two APE sensors have been implemented for the APE machine. The convert a light intensity into a pulse width modulated signal with the leading edge of the pulse synchronized with a reference clock. A sensor consists of a clock receiver, a switching transmitter, a synchronizing signal separator, a transmitter timing control, and a light-intensity-to-pulse-width converter, as shown in Figure 7.1. As in the case of the APEs, the output of the clock receiver is the composite clock signal. The synchronizing signal is separated from the composite clock signal by the separator. This synchronizing signal is used to trigger a one-shot circuit in the converter, whose timing network consists of a photocell and a capacitor. The width of the output pulse from the one-shot therefore depends on the light intensity which the photocell senses. The leading edge of the output pulse from the one-shot is in synchronism with the triggering signal, which is the synchronizing signal. The output of the one-shot, together with the clock signal and synchronizing signal, is sent to the transmitter timing control circuit, whose function is to turn the transmitter on and off at appropriate instants of time. All functional blocks in Figure 7.1 except the light intensity converter are identical to the corresponding parts in the APEs: Their details have been given in Chpater 6. The light intensity converter is shown in Figure 7.2. A COS/MOS AND gate and an inverter is connected to form the one-shot as shown. The output pulse width is depending on the product of the photoresistance and the capacitance.

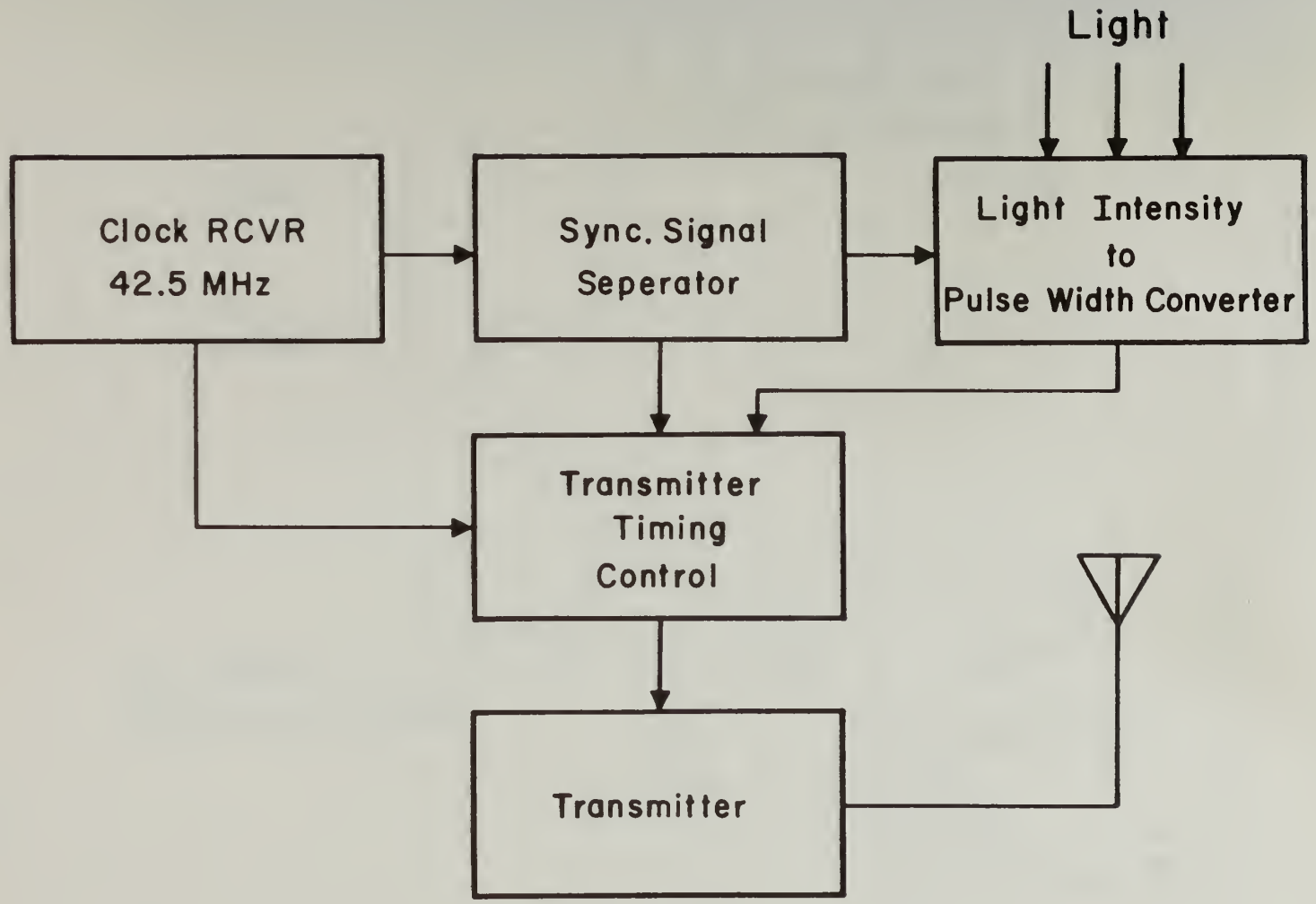


Figure 7.1 A Block Diagram of A Light Sensor

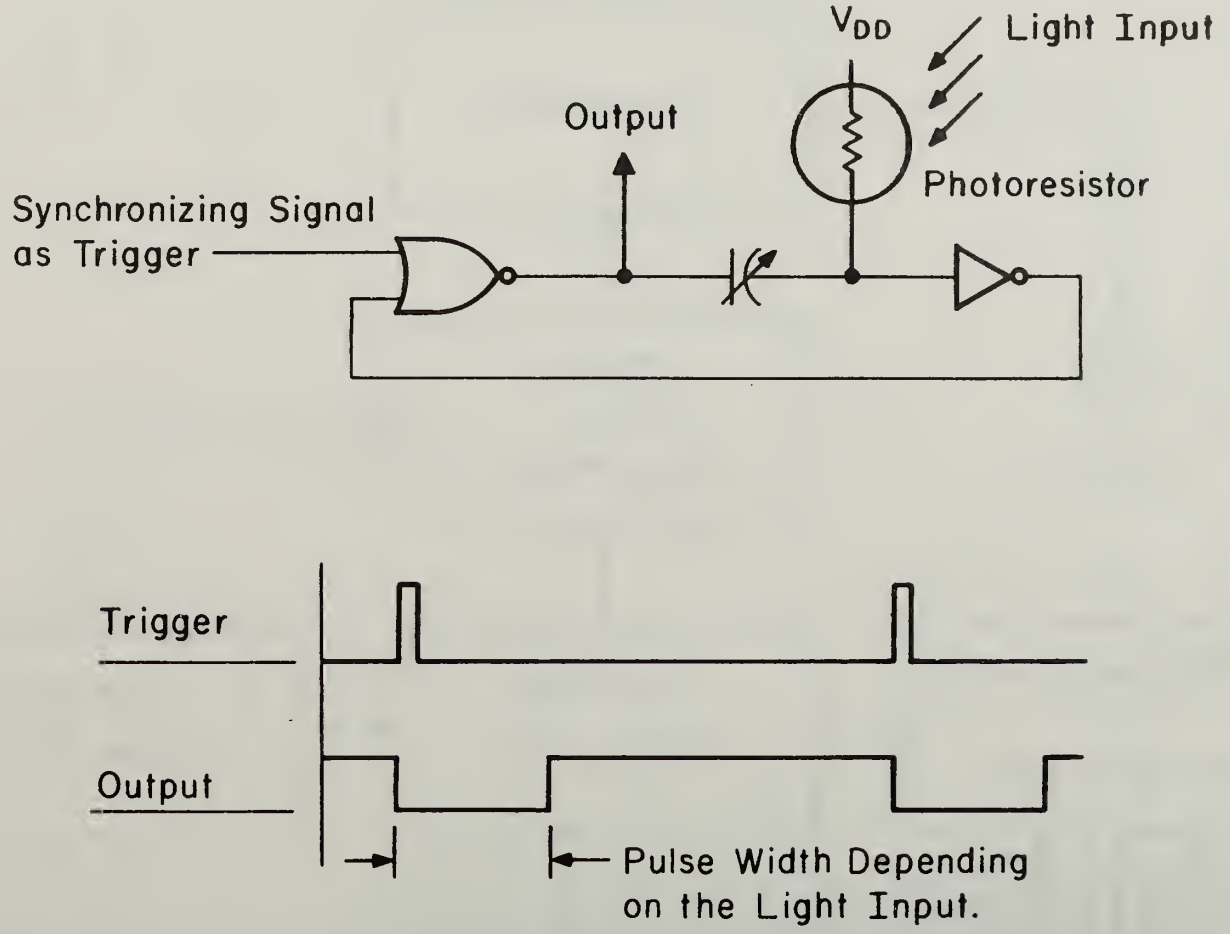


Figure 7.2 The Light Intensity to Pulse Width Converter

The former is dependent on the input light intensity. Consequently, the output pulse width is determined by the input light intensity.

7.2 The Remote Power Supply for the APE

The APE machine is equipped with a remote power supply for the APEs. The power is sent to the APE remotely to free it completely from any physical connection with other parts of the APE machine, therefore further enchancing the structural flexibility of the machine. The general requirements of the remote power source are: 1) The available power to the APEs placed in the active region of the power source should be about 100 mW. 2) The operation of the remote power supply must not interfere with the operations of the APE.

This part of the APE machine has been investigated originally by another graduate student in the Computer Hardware and Systems Research Group. The possibility of sending power to the APEs over a microwave frequency channel was first considered. A detail report on this study is given in reference (21). An experimental set-up having a 60-watt transmitter at 1296 MHz with a helical transmitting antenna was built. It was found that almost sufficient power for operating an APE could be delivered to an APE which is equipped with a quarter wave slotted line antenna for the reception of power and is placed a few feet from the transmitting antenna. However, it was also found that the transmission of the microwave power interferes with the operations of the APEs. As described in Chapter 5, each APE is equipped with three receivers operating at frequencies between 15,000 MHz and 42.5 MHz. The transistors employed in those receivers have necessarily a high gain-bandwidth product. Unfortunately, these transistors also response to RF signals at 1296 MHz. Without elaborate shielding and

feed-through filtering, the leakage of the 1296 MHz RF signal into the box housing the APE circuits is sufficient to upset the operations of the receivers. Some experiments have been conducted to house the APE circuitry inside an electromagnetically shielded box, with 1296 MHz resonant traps guarding all feed-through terminals feeding data signals into the APE circuits. With careful tuning of the 1296 MHz resonant traps, the interference could be reduced to an acceptable level. However, such tuning is quite critical and could be detuned easily by coupling with nearby objects. Hence, this approach is not employed for remotely powering the APEs.

The more suitable solution to the remote powering of the APE is by means of solar cells. This approach eliminates the harmful RF interference. These solar cells are constructed with a n-type silicon base material. A very thin layer of p-type material is formed on the n-type base through diffusion to produce a p-n junction with large area. When this p-n junction is short-circuited in darkness, no steady current will flow in the external circuit inspite of the existance of the contact potential of the p-n junction as expected on thermodynamic grounds. However, if light in a suitable range of wavelength is allowed to fall on the p-n junction, a voltage will develop across the external circuit and current starts to flow with a terminal characteristic as shown in Figure 7.3. More details about such a photovoltaic effect can be found in many good references (22) (23). Figure 7.4 shows the spectral response of the solar cells. For powering the APE, four solar modules, designated 5SM1020GE10PL by International Rectifier, are packed together on the top surface of the APE to absorb power from an array of incandescent lamps.

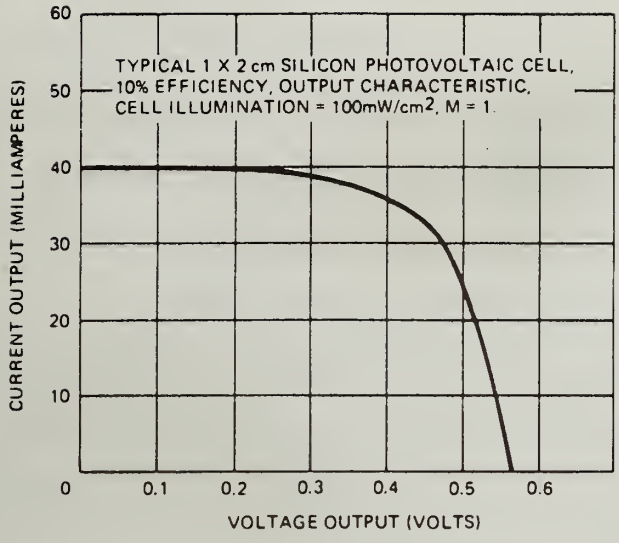


Figure 7.3 Typical Terminal Characteristics of a Photocell

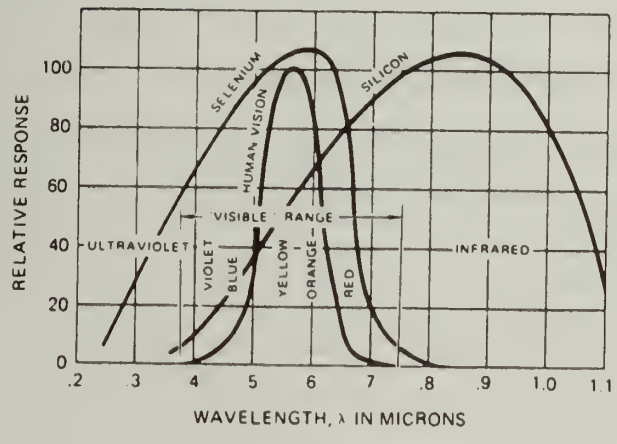


Figure 7.4 Spectral Response of Photocells

8. CONCLUSION AND OUTLOOK

With the help of the latest development in COS/MOS integrated circuits and high gain-bandwidth-product transistors, the APE machine has been successfully implemented. The maximum power consumption of an APE is actually about 70mW. The APE machine, as described in this thesis, has been operating satisfactorily. It is the world's first computer with variable topology of its kind. It is hoped that it will open up a new dimension in computer system and circuit design for highly flexible and reliable computers, with such advanced features as variable topology, incrementable computing power, readily mass-producible structure as well as self-checking and self-repairing capabilities.

In regard to the fault-tolerant features of the APE machine, only static checking procedures, such as the alive testing, have been implemented. Dynamic testing could easily be implemented by employing a highly reliable APE as the checker inside the control unit and comparing the output of the channel to be checked and that of the checker with identical inputs and operations. Furthermore, these checking procedures can be done automatically rather easily with the type of structure found in the APE machine: Whenever a failure is discovered in a specific channel, the element could be replaced automatically. If there are no more spare channels to replace the defective ones, the control unit could be readily programmed either to notify the operator or, in some inaccessible circumstances, to reconfigure the topology of the set of APEs such that the required number of APE channels is reduced to the available number, with some lower-priority functions of the program being cut out. To match the reliability of the other parts of the computer, information transmission of the APE machine should, of course, employ error checking and error correcting codes.

The APE machine described in this thesis made use of RF linkages to transmit data. There is obviously a limit to the number of APE channels that could be placed in a given frequency band. This limitation could be overcome with some sacrifice of the flexibility of the APE machine. For example, if information transmission is carried out over a confined space, such as a common bus, instead of the free space, then the number of APE channels could be increased by using several different confined spaces. In this case, special consideration must of course be given to the communication between the units associated with different confined space.

As an alternative approach to the design of the communication subsystem for the APE machine, a time-multiplexing scheme can be used instead of the frequency division scheme employed in the APE machine. In the time-multiplexing approach, the data receivers of all APEs are operated over the same fixed frequency channel having a much wider channel width. Then each type of APE is given a specific portion of the communication period to transmit the output data. To receive data from a specific type of APE. The data input register is programmed to take in data only during the specific portion of the communication period reserved for that type of APE. Such an approach eliminates the need of tuning the data input receivers and requires only one data receiver instead of two. On the other hand, additional logic circuits are required in this approach. Furthermore, it reduces the degree of homogeneity of circuits between different types of APE.

As described in Chapter 3, the fluctuation of the result obtained by stochastic processing would increase with the increase of the number of cascaded stages. This would limit the number of stages, and ultimately the number

of APEs, that could be used. This difficulty could not be overcome with stochastic computation using truly random SRPSs. Nevertheless, this fluctuation problem could be completely eliminated with special pseudo random SRPS whose period equals precisely the sampling integration period for the estimation of the mean value: Because the integration is taken over the entire period, it follows from the periodicity of the SRPS that the same result is obtained for every integration period. More importantly, the result thus obtained can be shown to be the mean value of the SRPS. In other words, the correct result is obtained every time without fluctuation.

LIST OF REFERENCES

1. Poppelbaum, W. J., "ONR Proposal N000 14-67-A-0305-0007", May 1969.

2. Ramamoorthy, C. V., "Fault-Tolerant Computing", IEEE Transaction on Computer, Vol. C-20, No. 11, November 1971.

3. Carter, W. C., and Bouricuis, W. G., "A Survey of Fault-Tolerant Computer Architecture and its Evaluation", Computer, January 1971.

4. Poppelbaum, W. J., et. al., "Stochastic Computing Elements and Systems", Fall Joint Computer Conference, Anaheim, California, 1967.

5. Ribeiro, S. T., "Random-Pulse Machine", IEEE Transaction on Electronic Computer, Vol. EC-16, June 1967.

6. Gains, B. R., et. al., "Stochastic Computing", AFIPS Proceedings, SJCC, Vol. 30, April 1967.

7. Gains, B. R., "Foundations of Stochastic Computing System", IEEE International Convention Digest, New York, March 1968.

8. Afuso, C., "Analog Computation with Random Pulse Sequences", Report No. 255, Department of Computer Science, University of Illinois, Urbana, Illinois, February 1968.

9. Poppelbaum, W. J., et. al., "Transformatrix - the World's Most Parallel Computer" to be published.

10. Esch, J. W., "Rascal - A Programmable Analog Computer Based on a Regular Array of Stochastic Computing Element Logic", Report No. 332, Department of Computer Science, University of Illinois, Urbana, Illinois, June 1969.

11. Esch, J. W., "A Display for Demonstrating Analog Computation with Random Pulse Sequence", Report No. 312, Department of Computer Science, University of Illinois, March 1969.

12. Korn, G. A., Random-Process Simulator and Measurement, McGraw-Hill New York, 1966.

13. Poppelbaum, W. J., Computer Hardware Theory, The Macmillan Company, New York, 1972.

14. Tauworthe, R. C., "Random Number Generation by Linear Recurrence Modulo Two", Mathematic of Computation, April 1965.

15. Sabolewski, J. S., et. al., "Pseudonoise with Arbitrary Amplitude Distribution", IEEE Transaction of Computers, Vol. C-21, No. 4, April 1972.

16. Galomb, S. W., Shift Register Sequence, Holden-Day, 1967.

17. Peterson, N. W., Error-Correcting Codes, MIT Press, Cambridge and Wiley, New York, 1961.

18. Cobbold, R. S. C., Theory and Applications of Field-Effect Transistors, Wiley-Interscience, New York, 1970.

19. Hoel, D. G., Introduction to Mathematical Statistics, John Wiley and Sons, Inc., 1971.

20. Breiman, L., Probability and Stochastic Processes, Houghton Mifflin Company, Boston, 1969.

21. Olson, D. L.,"Remote Power Supply for the APE System", Report No. 410, Department of Computer Science, University of Illinois, August 1970.

22. McKelvey, J. P., Solid-State and Semiconductor Physics, Harper and Row, New York, 1966.

23. Van Der Ziel, A., Solid State Physical Electronics, Prentice-Hall, Inc., 1968.

VITA

Yiu Kwan Wo was born on January 23, 1942 in Saigon, Viet Nam. He did his undergraduate work in Honors Electrical Engineering at McGill University, Montreal, Canada from 1963 to 1967. He was awarded the British Association Prize in 1964, and University Scholarships as well as the title of University Scholar for the academic years of 1964-65, 65-66, 66-67. In June 1967 he received the B. ENG. degree in Honors Electrical Engineering and was awarded the British Association Medal for high distinction in over all performance at McGill University.

From May to September 1965, he was a summer research assistant in the Pulp and Paper Research Institute of Canada in Montreal, working on the problem of automatic control of paper mills. The following summer, he spent four months as a summer research assistant in Whiteshell Nuclear Establishment in Manitoba, Canada, helping to design and implement electronic instruments associated with a particle accelerator. From May to September 1967, he joined the Canadian Marconi Company in Montreal as an engineer and took part in the development of an air-borne Doppler radar system.

In September 1967 he began his graduate studies under Professor Poppelbaum in the Computer Hardware and System Research Group, Department of Computer Science at the University of Illinois. He was awarded a University Fellowship from the University of Illinois for the academic years of 1967-68 and 1968-69. He has been employed as a Research Assistant by the Department of Computer Science since June 1969. He received his Master of Science degree in Electrical Engineering in February 1970.

He is a member of Phi Epsilon Alpha and Phi Kappa Phi.

Form AEC-427
(6/68)
AECM 3201

U. S. ATOMIC ENERGY COMMISSION

UNIVERSITY-TYPE CONTRACTOR'S RECOMMENDATION FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

AEC REPORT NO.	2. TITLE
COO-1469-0216 Report No. 556	APE MACHINE: A NOVEL STOCHASTIC COMPUTER BASED ON A SET OF AUTOMONOUS PROCESSING ELEMENTS

1. TYPE OF DOCUMENT (Check one):

- [X] a. Scientific and technical report
- [ ] b. Conference paper not to be published in a journal:
  - Title of conference _____
  - Date of conference _____
  - Exact location of conference _____
  - Sponsoring organization _____
- [ ] c. Other (Specify) _____

3. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

- [X] a. AEC's normal announcement and distribution procedures may be followed.
- [ ] b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
- [ ] c. Make no announcement or distrubution.

4. REASON FOR RECOMMENDED RESTRICTIONS:

5. SUBMITTED BY: NAME AND POSITION (Please print or type)

Yiu Kwan Wo
Research Assistant

Organization

Digital Computer Laboratory
University of Illinois
Urbana, Illinois 61801

Signature	Date
Y. K. Wo	February, 1973

FOR AEC USE ONLY

AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

-

6. PATENT CLEARANCE:

- [ ] a. AEC patent clearance has been granted by responsible AEC patent group.
- [ ] b. Report has been sent to responsible AEC patent group for clearance.
- [ ] c. Patent clearance not required.

IBLIOGRAPHIC DATA HEET	1. Report No. UIUCDCS-R-73-556	2.	3. Recipient's Accession No.

4. Title and Subtitle

PE MACHINE: A NOVEL STOCHASTIC COMPUTER BASED ON A SET OF AUTOMONOUS PROCESSING ELEMENTS

5. Report Date: February 1973

6.

7. Author(s)

iu Kwan Wo

8. Performing Organization Rept. No.

9. Performing Organization Name and Address

epartment of Electrical Engineering
niversity of Illinois
rbana, Illinois 61801

10. Project/Task/Work Unit No.

11. Contract/Grant No.

46-26-15-301

12. Sponsoring Organization Name and Address

S AEC Chicago Operations Office
800 South Cass Avenue
rgonne, Illinois 60439

13. Type of Report & Period Covered

Thesis Research

14.

15. Supplementary Notes

16. Abstracts

he APE Machine is a real-time computer with features such as reconfigureable structure, incrementable computing power and certain fault tolerant capabilities. This ighly flexible computer is dubbed APE Machine because its basic building blocks are set of Autonomous Processing Elements known as APEs. Each APE is a small processor n its own right. However, many APEs can be grouped together to form a more powerul processing unit. The APE Machine contains a set of sensors to perform input data cquisition and also a program control unit.

17. Key Words and Document Analysis. 17a. Descriptors

PE Autonomous Processing Elements

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages
	20. Security Class (This Page UNCLASSIFIED	22. Price